# Once Upon a Kernel: Extracting Important Events from Narratives

**Anshu Kiran Sharma, Miguel Castiblanco-Melendez, Alejandro Morales & Mark A. Finlayson**
Florida International University, Miami, FL, USA
{ashar076, mcast592, amora472, markaf}@fiu.edu

**Abstract**

Not all events in a narrative are created equal: some events are more important than others. *Kernel events*, a concept introduced in the field of narratology, are causally linked events that move the narrative forward, and cannot be removed without breaking the narrative's logical coherence. While event detection and extraction tasks have been widely studied in natural language processing and information retrieval fields, the idea of kernel events has been largely unexplored. In this work, we introduce the first corpus and model for kernel event detection. Our contributions include: the refinement of the kernel event concept captured in detailed annotation guidelines grounded in narratological principles; an annotation study yielding a gold-standard dataset of kernel events in narrative texts; and a first-of-its-kind kernel event detection system. Annotation achieved an inter-annotator agreement of 0.61 $\kappa$, underscoring the reliability of the guidelines. Using these data, we trained several models in both fine-tuned and generative modes for kernel event detection, with a LoRA fine-tuned Llama3 achieving an $F_1$ of 0.695. This work establishes a benchmark for kernel event detection, with potential applications in summarization, narrative similarity detection, and narrative understanding. We release our code and data for the benefit of other researchers.

**Keywords:** kernel events, narrative understanding, annotation guidelines, dataset creation, event detection, event extraction, fine-tuning, large language models, language resources

## 1. Introduction

Narratives are all around us, in a variety of modalities and genres, such as books, movies, and social media, or drama, comedy, and tragedy. Narrative is a crucial communicative mode that helps people make sense of our world. Narratives are distinguished from other discourse forms via their structural elements, which include events, plot, setting, and characters, all of which come together to form a world building representation that is communicated using a textual or audio-visual medium (Bal and Van Boheemen, 2017).

One notable characteristic of narratives is that they often remain recognizable across changes in modality, style, and genre: a single narrative can be retold or adapted and still remain recognizable as being the original narrative. For example, *Cinderella* exists both as a folktale in the Grimm Brothers' *Aschenputtel* (Grimm and Grimm, 1812/1988) as well as an animated movie in Disney's *Cinderella*. The Grimm version has a darker tone, featuring a magical tree rather than a fairy godmother and including graphic elements such as the stepsisters mutilating their feet to fit the glass slipper. In contrast, Disney's adaptation is a whimsical story with cute, talking animals. Despite these differences, core elements remain: an orphan mistreated by her stepfamily, the royal ball, a lost slipper, and a shoe-fitting test which, when completed by Cinderella, leads to a happy conclusion to her story.

Even within the same modality, a story can be retold with changes while maintaining recognizability. Marian Roalfe Cox's collection of Cinderella variants published in 1893 demonstrates the breadth of variation within the Cinderella narrative while highlighting its underlying structural consistency. While the Grimm Brothers' *Aschenputtel* and Charles Perrault's *Cendrillon* (Perrault, 1697) may differ in details such as a magical helper tree or a magical fairy godmother, they can still be perceived as versions of the same story. The existence of multiple versions of the same story raises a fundamental question: What constitutes the "sameness" of a story? How can narrative similarity be quantified? As we will discuss below (§3), these questions have occupied theorists of narrative for decades. Answering these questions has implications for various computational tasks, including narrative summarization, question answering, and story generation.

Several approaches have been proposed for measuring narrative similarity, including lexical similarity methods like *tf-idf* (Spärck Jones, 1972), word embeddings (Mikolov et al., 2013; Pennington et al., 2014), cosine similarity (Salton, 1989), or sentence embeddings like sentence transformers (Reimers and Gurevych, 2020). A different possible approach, however, attends to the structural factors that are central a narrative's identity: in particular, the narratological concept of **kernel events** describes the important events in a story that move it forward and cannot be deleted without completely ruining the story's logic. Two variants of a story should largely share the same kernel events. While extensive research has been conducted on event extraction (Grishman et al., 2005;



*Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 6009–6021
11–16 May 2026. 

Ji and Grishman, 2008; Chambers et al., 2014; Han et al., 2019; Zhong et al., 2024), the identification of kernel events remain essentially unexplored.

We introduce an annotation guide based on narratological principles, a gold-standard annotated dataset, and the first kernel event detection model trained and tested on that data. The paper is organized as follows. We start with a discussion of the important background concepts as well as related work (§2). We then explain how we refined the concept of kernel events by synthesizing a variety of theoretical perspectives (§3). Next, we discuss data collection and annotation procedure (§4), followed by the methods we explored to build the baseline system (§5). Finally, we conclude with a discussion of the results (§6) and the summary of our contributions (§7). All code and data have been archived to assist other researchers in building upon the work[1].

## 2. Related Work

### 2.1. Events

There have been a number of attempts to define the concept of event in general, both in narratological and computational fields. According to the narratologist Seymour Chatman (Chatman and Chilton, 1978), events are actions or happenings that bring about a change in the state; they are the building blocks of narratives involving agents or patients. Similarly, Mieke Bal (Bal and Van Boheemen, 2017) defines events as a process, a transition from one state to another state. Shlomith Rimmon-Kenan (Rimmon-Kenan, 1983), on the other hand, simply notes that events are things that happen, which can be denoted by a verb or an action name. While narratologists' theories are rich and subtle, they unfortunately have not been operationalized in actual linguistic annotations, making it a challenge to use them for computational work.

On the computational side, one of the standard sources of annotated event data is the Automatic Content Extraction 2005 Multilingual Training Corpus (ACE2005; Walker et al., 2006), which defines a lexically grounded theory of entities, relations, and events. ACE2005 comprises nearly 1800 annotated files of mixed genre text in English, Arabic, and Chinese. The corpus defines an event as a specific occurrence—either something that happens or a change of state—that involves participants. The annotation scheme lays out 8 event types: *life*, *movement*, *transaction*, *business*, *conflict*, *contact*, *personnel*, and *justice*, and 33 subtypes, as well as further identifying the event arguments (*agent*, *object*, *source*, and *target*) and attributes. ACE2005 recognizes only those events that fall into the named category, making it rather specific in its ontological commitments.

Early work focused on extracting ACE events used two consecutive classifiers, one for detecting triggers and the other for arguments (Ahn, 2006), achieving performances of roughly 0.223 $F_1$. Ji and Grishman (2008) advanced the prior state of the art by using the full document context as well as cluster-wide statistics about the frequency of trigger and argument. That work achieved 0.673 $F_1$. More recent approaches have explored feed-forward neural networks, including deep approaches; for example, Li et al. (2013) used multi-layer neural nets to extract event triggers and arguments with an $F_1$ score of 0.704, while Wadden et al. (2019) used fine-tuned BERT embeddings with text spans to capture both local and global contexts, achieving an $F_1$ of 0.697. Huang and Peng (2021) used a similar method employed by (Wadden et al., 2019) but expanded the event coreference module by incorporating deep value networks into the base model to achieve 0.691 $F_1$.

In the generative line, Du and Cardie (2020) have formulated the problem as a question-answering task, achieving 0.724 $F_1$. Lu et al. (2023) further explored question generation and answering formulations for event extraction, reporting 0.76 $F_1$. Shi et al. (2023) proposed a hybrid detection and generation framework with separate encoders for event and event argument extraction, achieving 0.79 $F_1$ for event extraction. Zhong et al. (2024) uses a two-phase graph inference network with the first phase of document-level graph inference to get document context and a second phase of an LSTM-based approach to achieve 0.822 $F_1$. Zhu et al. (2024) leverage the extraction ability of small language models as well as the instruction following ability of LLMs to improve event extraction results with an average improvement of 2.7%.

The other major computational approach to events and temporal information, one more linguistically grounded, is TimeML (Pustejovsky et al., 2003), (Saurí et al., 2006). TimeML marks four types of temporal information: temporal signals, temporal expressions, events, and temporal relationships. TimeBank was annotated with the TimeML schema and contains 183 English news articles with more than 27,000 event and temporal annotations and 61,000 non-punctuated tokens. TimeML defines events as situations that happen or occur. They also allow states or circumstances in which something obtains or holds true to be events. TimeML, on the other hand, accepts all events in a text as one of the 8 classes of events (*occurrence*, *reporting*, *perception*, *aspectual*, *intensional action*, *state*, *intensional state*, and *unknown*).

There are a number of systems for performing

[1]Code and data can be downloaded from https://doi.org/10.34703/gzx1-9v95/KBJBW4

TimeML event extraction, including TARSQI (Verhagen and Pustejovsky, 2012), CAEVO (Chambers et al., 2014), and ClearTK (Bethard, 2013). TARSQI uses EVITA (Events In Text Analyzer; Saurí et al., 2005), which applies linguistically motivated rules along with disambiguation using Bayesian classifier, achieving a score of 0.8012 $F_1$. CAEVO runs on sieve-based architecture that uses Navy-Time to extract events with 0.803 $F_1$. ClearTK achieved 0.773 $F_1$ on TimeBank data using models with CRFs, SVMs, and logistic regression, and the best classifier was selected using grid search over classifiers and parameter settings. Arnulphy et al. (2015) have used a combination of CRFs and kNNs for TimeML event detection in English and French and achieved an $F_1$ score of 0.86. Prabhu et al. (2019) have developed a deep learning model ALINED to extract TimeML events in English, Spanish, Italian, and French with 0.827 $F_1$. Han et al. (2019) developed a model using BERT embedding and BiLSTM layers which has an RNN scoring function with a measure of 0.909 $F_1$.

Compared to ACE—which focuses on identifying events and their associated entities in real-world, domain-specific corpora such as newswire, broadcast news, and conversational speech—TimeML offers a more general and linguistically grounded framework. While ACE emphasizes entity-event relations within a fixed ontology, TimeML prioritizes the temporal structure and semantics of events, capturing nuances such as temporal relations and expressions. Given TimeML's broader coverage and greater sensitivity to linguistic and semantic detail, we adopt it over ACE2005 as a more suitable annotation scheme for our task.

## 2.2. Fabula and the Plot

Narratives are comprised of events, but there is often a difference between the order of events in the "story world" and the order in which those events are presented in the narrative discourse. This distinction has been much discussed in narratology. For instance, the Russian Formalist school called these *fabula* and *syuzhet* (Shklovsky, 1990; Tomashevsky, 1965), where *fabula* is the series of events in the world of the story, while *syuzhet* is the presentation of those events in the narrative form. Tomashevsky described the difference as the fabula being the actual actions of the characters, while the syuzhet or plot is how the audience gains knowledge of it. In the French Structuralist tradition, on the other hand, Gérard Genette (1976) called the sequence of events *historie* [story] and the presentation of the events in a discourse *récit* [narrative]. According to him, *historie* is just the content of the narrative regardless of how dramatic or how complete the events of the incident being described are while *récit* is the actual text of the narrative. Within recent structuralism and narratology, Jonathan Culler (2004) names these elements *story*—a sequence of actions or events independent of how they appear in the discourse and *discourse*, the discursive presentation or narration of those events. Bal and Van Boheemen (2017) defines *story* as the content of that text and produces a particular manifestation of a fabula while a *fabula*, according to her, is a series of logically and chronologically related events that are caused or experienced by actors.

In this work, we have adopted the terms *fabula* and the *plot* as they are commonly used in recent narratological theory. In alignment with Bal and Culler, we use **fabula** to refer to the sequence of events that happen in the story world, the world may be real or fictional. The audience of the narrative builds a world model where these events take place in a particular order. On the other hand, a **plot** is an arrangements of the fabula into a sequence, as presented in the discourse.

## 2.3. Kernel Events

Not all events in a narrative contribute equally to the logic and continuation of the plot. Some events are logically central, while other events serve to flesh out characters or create an atmosphere. The distinction between these different types of events has been studied extensively in narratology by scholars such as Tomashevsky, Vladimir Propp, Roland Barthes, and Seymour Chatman.

Propp's structural analysis of Russian folktales in *Morphology of the Folktale* (Propp, 1928/1968) was one of the earliest investigations of the idea that certain events in narratives are privileged. Propp analyzed a corpus of 100 Russian hero tales and identified 31 narratives *functions* that define the underlying common structure of those tales. Functions represent the essential components of a story's plot, and include elements such as *Villainy*, *Departure* (the hero leaves home), *Struggle* (the hero and the villain join in direct combat), *Victory* and *Recognition*. He suggests that folktales, despite variations in actual detail, maintain a core identity through the presence of these core functions, and these functions shape a narrative's progression. Barthes (1975) built upon Propp's work to propose that events in a narrative could be split into important events—which he referred to as *cardinal functions* or *nuclei*—and disposable events—which he called *satellites*. According to Barthes, nuclei events are "hinges" which either open or close an uncertainty in the narrative.

Tomashevsky (1965) also noted that not all events are as important for the forward movement of a narrative, some events are more necessary than others. He called these events *bound motifs*, and proposed that the relative importance of the

motifs to a story can be determined by retelling the story and comparing the abridged version with the original narrative. The bound motifs are the ones that cannot be deleted without fully disturbing the whole chronological and causal chain of events.

Chatman (1978) introduced the term *kernel events*, expanding on Barthes' ideas. Like Barthes, Chatman insisted that kernel events are "nodes" or "hinges" which force the movement of the narrative into at least one of the multiple branching paths. Kernel events that come later in the text are the direct consequences of the kernel events that came before them.

A key idea running through all theories of kernel events in the narratological literature is the notion that some events naturally follow on others, via a cause-effect relationship. This sensitivity to causal effects appeals to a very deep, yet challenging, set of relevant philosophical criteria. According to David Hume (1902), "*we may define a cause to be an object, followed by another, and where all the objects similar to the first are followed by objects similar to the second. Or in other words where, if the first object had not been, the second never had existed.*". Based on the second definition by Hume, David Lewis (1973) proposed a counterfactual theory of causation which can be summarized as *if the causing event had not occurred, the caused event would not have occurred either*. That is to say, whether the caused event occurs or not is dependent on the causing event to occur. Lewis's definition requires the events to be distinct from one another such that events are not a part of each other. Hobbs (2005) builds on these foundations by distinguishing between causal chains, linear sequences where each event depends on the previous one, and causal complexes, the complete sets of relevant conditions sufficient for an effect. Within his framework, intermediate links in a chain may themselves be described as causes, but only relative to a smaller complex nested within the larger one. This observation is crucial for our purposes. It shows that while intermediate events are causally relevant, their status as causes is derivative of the broader complex in which they are embedded. Accordingly, we treat kernel events as the initiating and terminating points of a causal chain.

Prior work in cognitive psychology has also examined how readers identify structurally important events in narratives. Research on causal network models of story comprehension shows that events with greater causal connectivity are perceived as more central and are more likely to be recalled (Trabasso and Sperry, 1985; Graesser et al., 1994). Similarly, related work on event segmentation theory (Zacks and Swallow, 2007) argues that humans organize experience into discrete event units based on changes in situational context. These findings provide support for the notion that narratives contain events that are more central to understanding than others, similar to the concept of kernel events.

There has only been limited work on computational approaches to kernel events and related ideas. Otake et al. (2020) tried to determine event salience in a small dataset of Russian folktales by deleting events and computing and comparing the coherence score with the original text, achieving a 0.3 MAP score, where salience estimation is treated as a ranking problem. Wilmot and Keller (2021) expanded this work to plays and novels using a transformer language model with a 0.319 MAP. Their work only tackles a single dimension of kernel events, i.e., textual coherence. While this provides a measure of how well coherence is preserved, it does not capture the broader set of factors that we consider in characterizing kernel events.

## 3. Refining the Concept of Kernel Events

As can be seen above, there are number of different approaches to conceptualizing events that are central to a narrative. Tomashevsky's approach to identifying bound motifs involves abridging and retelling the narrative, and while he acknowledges the role of these motifs in the causal chain of events, he does not explore in detail how they contribute to the construction of these chains. On the other hand, Barthes asserts that nuclei must be both "consecutive and consequential". For Barthes, what defines a nucleus is its role as a pivotal moment involving uncertainty and risk, regardless of how engaging the event may be. While much of the later work on cardinal functions builds on Barthes' concept, his framework remains ambiguous regarding the formalization of the consequential and consecutive nature of nuclei.

While Tomashevsky was among the first to conceptualize important events in a narrative, most subsequent work builds on Barthes, as Tomashevsky's analysis remains limited to their chronological and causal roles. Barthes describes these key events as narrative hinges, a concept further developed by Chatman, who argues that they direct the narrative along available paths. While both Barthes and Tomashevsky emphasize the sequential nature of important events, Chatman extends this by highlighting the cause-and-effect relationship between kernel nodes, though without deeply exploring causality. Contrasting with other theories, Chatman elaborates on the causal relationship between kernel events, and gives a good starting point for detecting kernel events, where he mentions that kernel events "raise and answer questions," where the answer the question raised by one kernel event is another kernel event.

Despite these various contributions, the formalization of kernel events remains incomplete, in that we lack precise guidelines for identifying kernel events. Therefore we must refine the theory before we can hope to create reliable annotated data. Since Chatman's work remains the most comprehensive of all with regard to the elaboration of the causal relationships between events, we start with the idea of **kernel events** as described by Chatman: Kernel events are those events that either start or close an open possibility in a plot and move the plot forward by opening new paths for the plot to move towards. Finally, each kernel event should hold a "chrono-logical" relationship (Chatman's phrasing) with the kernel events that come before or after it.

For example, in the story of Cinderella, her dropping the shoe is a kernel event as it propels the story by triggering the Prince's shoe test. Without the shoe drop, there would be no shoe test and thus no story. Similarly, the Prince finding the shoe is another kernel event: if the Prince had never found the shoe, the shoe test would have never happened, causing the story to end right there.

Every kernel event is a direct consequence of the kernel events that came before it, and is the cause of the kernel events that occur after it, which is to say that a causal connection exists between kernel events. Taking the example of Cinderella again, the shoe test is a direct effect of Cinderella dropping her shoe, while her going to the ball in the first place is the cause of her dropping the shoe.

## 4. Data and Annotation

### 4.1. Collection of Texts

Narratives exist in a wide range of forms such as novels, plays, and movies. While some narratives have highly complex and non-linear plots, others are more straightforward. For our work, classical fairy tales were selected as the primary narrative form due to their simple and linear narrative structure, which allows for easier identification of causal relationships between events.

Fairytales typically follow a straightforward linear chronological order: an introduction, followed by a series of events leading to the climax, which in turns directly leads to the conclusion of the tale. This structure makes it easier to identify kernel events, and is an appropriate simplification for this first-of-its-kind study.

To eventually allow us to explore the hypothesis that two narratives can be considered similar if they share similar kernel events, the dataset includes multiple versions of the same folktales. For annotation, we collected a dataset of 50 folktales. Of these, 10 are Russian folktales drawn from Vladimir Propp's Morphology of the Folktale, as the feasibility of event identification has been demonstrated in the ProppLearner corpus (Finlayson, 2015). The remaining 40 include 15 standalone folktales and 25 versions of well-known fairytales. This distribution is summarized in Table 1.

| Story Type | # text | # tok. | # evt. | # ker. |
|---|---|---|---|---|
| Standalone tales | 15 | 22,354 | 3,283 | 1,025 |
| Propp's tales | 10 | 14,513 | 2,206 | 702 |
| Cinderella | 5 | 13,165 | 1,900 | 538 |
| Hansel and Gretel | 5 | 10,680 | 1,513 | 430 |
| Red Riding Hood | 5 | 5,685 | 772 | 221 |
| Beauty and the Beast | 4 | 17,418 | 2,672 | 444 |
| Jack and the Beanstalk | 3 | 6,312 | 866 | 205 |
| Emperor's New Clothes | 3 | 3,709 | 484 | 120 |
| Total | 50 | 93,836 | 13,696 | 3,685 |

Table 1: Distribution of data in the collected pilot dataset. Here, #tok. = number of tokens, #evt. = number of events, #ker. = number of kernel events in the dataset

### 4.2. Annotation

We consolidated key ideas from narratology to develop an annotation guide for identifying events and kernel events in narrative texts. The guide provided clear instructions on how to distinguish kernel events, emphasizing the causal relationship between them.

To identify kernel events in the stories, the annotators followed the steps given below:

1. Read carefully through the given text.
2. Identify the climax of the text.
3. Identify the kernel events that comprise the climax.
4. Start from the beginning and mark the events that lead up to the climax event.
5. Finally, mark the events that are the consequence of the climax event

We adopted a simplified TimeML scheme marking only the event headword, defined as the main non-auxiliary verb conveying the phrase's core meaning and grammatical or syntactic function. The datasets used in this work follow a compatible TimeML annotation variant in which each event span is associated with an explicit head index[2], allowing evaluation to be conducted consistently at the headword level.

The following example illustrates an annotation where events are single underlined, while kernel events are double underlined.

[2] https://projects.csail.mit.edu/workbench/update/guides/03%20-%20Events_v2.0.0.pdf

*He obliged Cinderella to sit down, and, putting the slipper to her little foot, he found it went on very easily, and fitted her as if it had been made of wax.*

A team of five annotators carried out the annotation. All annotators were undergraduate research assistants affiliated with our laboratory and were specifically trained for this task by the first author. Annotations were carried out using the *Prodigy* (Montani and Honnibal, 2017) tool. We trained the annotators in stages with them initially annotating general events in the narrative texts to develop consistency in event identification. Once they demonstrated reliability in this task, they moved on to annotating kernel events. Once they were comfortable with both types of annotations, they annotated both events and kernel events in parallel. After they reached a sufficiently high agreement score for event-only annotation, the annotators annotated every story once with random checks to monitor consistency and agreement.

After each round of annotation, we calculated inter-annotator agreement scores and conducted adjudication, led by the first author. During adjudication, annotators reviewed discrepancies in their annotations and resolved disagreements to establish a gold-standard dataset, guided by the first author. Since annotators worked independently during annotation and could only refer to the annotation guide, these meetings allowed us to refine the guidelines based on observed inconsistencies and discussion insights. Each revision of the annotation guide incorporated lessons learned from adjudication, improving clarity and reducing ambiguity in future rounds of annotation. The final gold-standard annotations were produced through this adjudication process rather than by selecting a single annotator as ground truth.

### 4.3. Inter-Annotator Agreement

To calculate the inter-annotator agreement score, we adopted the use of the average pairwise $F_1$ as an inter-annotator agreement score, where one annotation is randomly chosen as the gold standard and the other is treated as the prediction. For event annotation, only two measures the strict $F_1$ and partial or graded $F_1$ were calculated. For the strict $F_1$, the annotations were given a score of 1 for the number of the same tokens only if the gold and the predicted event spans matched exactly, otherwise, they were not counted. But for partial $F_1$, matching event spans were weighted from 0 to 1 based on the number of shared tokens (using the Jaccard score (Jaccard, 1901)), and the number of the same tokens was calculated based on the sum of the grades. For kernel event annotation, we also computed Cohen's Kappa ($\kappa$) along with the strict as well as partial $F_1$. The results for the annotation tasks are given in Table 2. $\kappa$ of 0.611 signifies moderate agreement (McHugh, 2012).

| Annotation Type | Metric | Score |
|---|---|---|
| **Events Only** | Partial Match $F_1$ | 0.839 |
| | Exact Match $F_1$ | 0.834 |
| **Kernel Events** | Partial Match $F_1$ | 0.691 |
| | Exact Match $F_1$ | 0.689 |
| | $\kappa$ | 0.611 |

Table 2: Inter-annotator agreement score for Events and Kernel Event Annotation

## 5. Approach

We used our annotated data to develop models for both general and kernel event extraction in narrative texts. The approach consists of data preprocessing, fine-tuning strategies, generative strategies, and evaluation across multiple event extraction datasets. Our objective was to train models capable of distinguishing both general and kernel events in folktales. We also assessed the generalization of general event detection from folktales to news data. All models trained and fine-tuned in this work are open-weight, if not fully open-source, to support reproducibility and advance open science.

### 5.1. Preprocessing

To prepare the annotated dataset for model training, we first split the text into individual sentences. Since narrative structures could span multiple sentences, a simple sentence-level approach would result in the loss of contextual information. To address this, we implemented a sliding window technique with a size of 512 tokens, ensuring that the context within a given segment remained sufficiently broad while staying within the input length constraints of transformer-based models.

We trained the models for two distinct subtasks: (1) extracting all events present in a narrative, and (2) identifying kernel events, which are a subset of general events.

For kernel event extraction, only the Fairytale dataset was used. But for general event extraction, we supplemented our dataset with the *TimeBank* corpus, a widely used dataset for event annotation in news texts. This allowed us to test the transferability of our models by evaluating those trained on our Fairytale dataset against TimeBank. We further tested the models on additional event extraction benchmarks, including the *N2* corpus (Finlayson et al., 2014), and Propp's annotated folktales from *ProppLearner* corpus (Finlayson, 2015), assessing their ability to generalize across different narrative

and non-narrative domains.

Following preprocessing, we split the dataset into training, validation, and test sets using an 80-10-10 ratio. All datasets used were partitioned separately using this ratio, ensuring that models trained on a specific dataset were evaluated on distinct validation and test sets drawn from the same source.

### 5.2. Fine-Tuned Models

To establish a baseline for event extraction, we fine-tuned several transformer-based models with open weights, ensuring accessibility and reproducibility. Specifically, we trained BERT (*bert-base-cased*) (Devlin et al., 2019), RoBERTa (*roberta-base*) (Liu et al., 2019), and T5 (*t5-small*) (Raffel et al., 2020). These models have been widely used in event extraction tasks and serve as a strong foundation for identifying both general and kernel events. BERT and RoBERTa, as encoder-only models, excel in token classification tasks like event detection, while T5's encoder-decoder structure enables the formulation of event extraction as a text-to-text generation task, providing additional flexibility. Including both architectures allowed us to compare their relative strengths for our specific extraction tasks.

All models were trained on an Nvidia A100 GPU, leveraging standard training frameworks. For BERT, RoBERTa, and T5, we used the Hugging Face *transformers* library (Wolf et al., 2020), which provided pre-trained weights and streamlined fine-tuning procedures. Each model was trained for exactly 5 epochs, requiring approximately half an hour per model run.

Event extraction was treated as a binary token classification task. Each token was assigned one of two labels: EVENT or NON-EVENT. The models, therefore, learned to directly predict whether a given token represented an event. Recall, we only annotated headwords; thus, only one token per event expression was labeled EVENT, unlike the original TimeML scheme. We used the same setup for kernel event extraction, but with the labels restricted to distinguishing kernel events from all other tokens.

### 5.3. Generative Models

For larger generative models, we fine-tuned Llama3 (*8B*) (AI@Meta, 2024) and Mistral (*7b*) (Jiang et al., 2023) as well as models with a larger context window, Llama 3.2 (1B and 3B) using Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA) (Hu et al., 2021). LoRA allowed us to update only a small subset of trainable parameters while keeping the pre-trained weights frozen, significantly reducing memory requirements. Consistent with prior studies on LoRA fine-tuning of Llama-based models (Hu et al., 2021; Zhang et al., 2025; Greenewald et al., 2025; Mu et al., 2025), we applied LoRA with a rank of 32. All models were trained for 5 epochs.

These models were selected due to their improved handling of longer text dependencies, making them well-suited for kernel event extraction, where causal relationships across the narrative play a central role. To optimize performance, we also applied prompt engineering techniques, ensuring that the models generated outputs aligned with event extraction objectives.

Again, we used Nvidia A100 while fine-tuning Llama3, and Mistral and it was conducted using the *unsloth* library (Daniel Han and team, 2023), which provides optimizations for large-scale model training. Each model required approximately an hour to fine-tune.

For the generative models, we framed event extraction as a supervised text-to-text task. Each training instance consisted of a prompt containing task instructions, the narrative context, and the corresponding gold list of kernel events (or events). We fine-tuned the models to generate the event lists, aligning their outputs with the annotated tokens. At evaluation time, we gave the models only the prompt and narrative text, and we compared their generated lists of events or kernels against the gold annotations to measure performance. The best performing prompts for fine-tuning and evaluation are documented in Appendix A.

## 6. Results and Discussion

We evaluated cross-domain generalization by testing models trained on structured narratives (Fairytales) against news text (TimeBank), and vice-versa, to assess whether existing event extraction datasets—primarily developed for news texts—transfer effectively to narratives. We report performance using $F_1$ score and analyze how various preprocessing techniques and cross-domain evaluation impact event extraction generalizability.

Table 3 shows RoBERTa achieves best general event extraction ($F_1$=0.93) while Llama3 excels at kernel extraction ($F_1$=0.695). The lower score compared to general event extraction highlights the challenge of identifying kernel events, which are sparse in texts and require deeper contextual understanding. Kernel events represent a subset of all events and are sparsely distributed throughout a narrative. This imbalance contributed to lower model performance, as fewer training examples were available for kernel event identification. For completeness, we also tested GPT-4.1 and Claude Sonnet 4.5 in a zero-shot setting, both yielding identical scores ($F_1$ = 0.35) fully driven by cases with no kernel events, indicating general LLMs fail to extract kernel events without task-specific tuning. The same

| | Event - Timebank | | | Event - Fairytale | | | Kernel - Fairytale | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| BERT | 0.878 | 0.889 | **0.884** | 0.921 | 0.923 | 0.922 | 0.522 | 0.440 | 0.478 |
| BERT sliding window | 0.799 | 0.867 | 0.832 | 0.849 | 0.877 | 0.863 | 0.555 | 0.172 | 0.262 |
| RoBERTA | 0.848 | 0.907 | 0.877 | 0.937 | 0.934 | **0.935** | 0.588 | 0.294 | 0.392 |
| RoBERTA sliding window | 0.801 | 0.901 | 0.848 | 0.873 | 0.862 | 0.868 | 0.489 | 0.177 | 0.260 |
| T5 | 0.875 | 0.806 | 0.839 | 0.888 | 0.720 | 0.795 | 0.677 | 0.509 | 0.581 |
| T5 sliding window | 0.794 | 0.355 | 0.490 | 0.927 | 0.343 | 0.501 | 0.621 | 0.358 | 0.454 |
| **Llama3** | 0.806 | 0.802 | 0.788 | 0.881 | 0.859 | 0.858 | 0.708 | 0.700 | **0.695** |
| Llama3 sliding window | 0.814 | 0.809 | 0.799 | 0.887 | 0.805 | 0.833 | 0.442 | 0.459 | 0.407 |
| Mistral | 0.785 | 0.817 | 0.777 | 0.842 | 0.834 | 0.823 | 0.613 | 0.614 | 0.599 |
| Mistral sliding window | 0.740 | 0.764 | 0.737 | 0.852 | 0.788 | 0.812 | 0.405 | 0.439 | 0.377 |
| Llama 3.2 1B | 0.767 | 0.809 | 0.762 | 0.834 | 0.796 | 0.796 | 0.565 | 0.546 | 0.540 |
| Llama 3.2 1B sliding window | 0.715 | 0.770 | 0.726 | 0.775 | 0.649 | 0.690 | 0.301 | 0.378 | 0.309 |
| Llama 3.2 3B | 0.809 | 0.816 | 0.791 | 0.779 | 0.844 | 0.792 | 0.599 | 0.592 | 0.579 |
| Llama 3.2 3B sliding window | 0.771 | 0.775 | 0.762 | 0.821 | 0.723 | 0.756 | 0.328 | 0.423 | 0.341 |

Table 3: Kernel event and event-only extraction results. Underlined bold indicates best $F_1$ per dataset.

| **Dataset/Models** | | **Fairytale** | | | **TimeBank** | | | **Propp** | | | **N2** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| BERT | Fairytale | 0.92 | 0.92 | 0.92 | 0.83 | 0.72 | 0.77 | 0.94 | 0.69 | 0.80 | 0.83 | 0.77 | 0.80 |
| | Timebank | 0.92 | 0.79 | 0.85 | 0.88 | 0.89 | 0.88 | 0.94 | 0.64 | 0.76 | 0.83 | 0.74 | 0.78 |
| | Propp | 0.79 | 0.89 | 0.83 | 0.73 | 0.76 | 0.74 | 0.89 | 0.92 | 0.90 | 0.66 | 0.81 | 0.73 |
| | N2 | 0.84 | 0.85 | 0.84 | 0.75 | 0.84 | 0.79 | 0.90 | 0.69 | 0.78 | 0.91 | 0.85 | 0.88 |
| RoBERTa | Fairytale | 0.94 | 0.93 | 0.94 | 0.82 | 0.76 | 0.79 | 0.93 | 0.70 | 0.80 | 0.82 | 0.79 | 0.80 |
| | Timebank | 0.89 | 0.83 | 0.86 | 0.85 | 0.91 | 0.88 | 0.94 | 0.64 | 0.76 | 0.87 | 0.76 | 0.81 |
| | Propp | 0.84 | 0.87 | 0.86 | 0.77 | 0.68 | 0.72 | 0.89 | 0.90 | 0.90 | 0.70 | 0.80 | 0.75 |
| | N2 | 0.84 | 0.85 | 0.85 | 0.77 | 0.82 | 0.80 | 0.90 | 0.70 | 0.79 | 0.91 | 0.88 | 0.89 |

Table 4: Precision, Recall, and $F_1$ scores for BERT (top) and RoBERTa (bottom) when trained on each source dataset (rows) and evaluated on each target dataset (columns).

prompt used for fine tuning the models was used for inference and provided in Appendix A.

Sentence splitting produced the best results among preprocessing strategies, preserving event boundaries and narrative structure. The sliding-window method performed worse, as it truncated sentences and dispersed event context, reducing coherence. Since kernel events depend on surrounding context and transformer-based models are trained on full sentences, fragmentation likely caused the performance drop.

For general event extraction, the best-performing model trained on TimeBank achieved an $F_1$ of 0.884 using BERT. However, when tested across datasets as shown in Table 4, performance dropped significantly. The best model trained on the fairytale dataset and tested on TimeBank achieved an $F_1$ of 0.79, while the best model trained on TimeBank and tested on fairytale yielded an $F_1$ of 0.85. The drop in performance underscores the importance of domain-specific training for event extraction tasks.

The sharp drop in performance of Fairytale-trained models on TimeBank reveals the limits of cross-domain event extraction. News events in TimeBank follow different structural patterns than the causal, logic-driven events in fairytales, highligting the need for domain-specific training.

These results demonstrate that while existing event extraction models perform well within their respective domains, kernel event extraction remains a challenging task due to the sparse relevant examples and the need for deeper contextual modeling.

Direct comparison with recent event extraction architectures is not straightforward because those models are designed to identify ACE or TimeML events rather than causally central events in a narrative. Kernel event detection introduces an additional layer of abstraction beyond standard event extraction. Accordingly, our experiments focus on establishing the first baselines across various machine learning architectures. We view this work as defining the task and benchmark upon which future model development can build.

### 6.1. Error Analysis

To better understand the limitations, we conducted an error analysis of the best-performing model on the 412-sentence test set. The analysis reveals

perfect predictions in 49% of sentences, with errors comprising: complete misses (17%), false positives (11%), under-prediction (9%), over-prediction (6%), partial overlap (5%), and wrong predictions (3%).

In over-prediction, the model correctly identified the gold kernel event but added extra verbs. For example, in sentence, "*But Beauty soon recovered her fright, for Beast having said, in a mournful voice, "then farewell, Beauty," left the room.*", the gold event was *left*, but the model predicted *{'recovered', 'said', 'left'}*, reflecting over generalization to high-frequency reporting verbs such as *said* or *replied*.

Under-prediction occurs mainly in cases of multi-clause sentences, such as "*One day, she called them all together and said, "Dear children, I must go out to find some food.*", where the model captured *go*, but missed *said*. Partial overlap cases involved near-synonyms like *called* instead of *said* while false positives arose in sentences without kernel events, especially when generic action verbs like *went, saw, looked* appeared.

These findings indicate that the model captures surface-level verb cues effectively but fails to distinguish causally central events to the story. Future work should focus on improving model architectures to better capture the causal dependencies and narrative centrality of kernel events.

## 7. Contributions

This work introduces the first computational framework for kernel event extraction, leveraging the insights gained from narratology for automatic identification. We contribute a gold-standard dataset of 50 fairytales with 3,685 kernel and 13,696 TimeML events, establishing the first benchmark for kernel event extraction and expanding TimeML beyond its predominant use in news data.

To support the annotation process, we developed a comprehensive annotation guide that provides clear criteria for identifying kernel events. This guide not only facilitates consistent annotation but also serves as a reference for future research in narrative event extraction. Furthermore, we introduce the first-ever kernel event extraction model, providing a computational approach to automatically identifying these essential narrative elements. Evaluation across 14 model configurations establishes baseline performance ($F_1$=0.695) and reveals a fundamental challenge for future research.

All datasets, annotation guidelines, and models are made publicly available to ensure reproducibility and to support further research in this area[3].

[3]Code and data can be downloaded from https://doi.org/10.34703/gzx1-9v95/KBJBW4

## 8. Limitations

Kernel events do not exist in isolation. Each kernel event is an effect of kernel events that came before it and a cause of kernel events that come after. This interconnection means that accurately identifying kernel events requires understanding the full context of the narrative rather than each event in isolation.

Our current approach primarily focuses on identifying kernel events at the sentence level, but it does not fully incorporate the context of the entire text. Since kernel events' significance comes from their role in the overall story progression, a more comprehensive approach would need to model event dependencies across the full narrative structure. Addressing this limitation will be the focus of our future work. We aim to develop methods that leverage document-level context to improve kernel event identification, ensuring that extracted events are not only structurally significant but also correctly positioned within the story's causal framework.

In addition, the dataset mainly consists of fairytales, which are relatively linear and causally explicit. Performance will likely differ on narratives with more complex or non-linear structures. Furthermore, while the annotation guidelines aim to operationalize causal centrality, borderline cases may introduce annotator subjectivity. Future work should evaluate kernel event identification across more diverse narrative domains.

## 9. Ethics Statement

This study adheres to ethical research practices by utilizing publicly available narrative texts to construct the dataset. While most stories are drawn from public-domain sources, some texts are subject to copyright restrictions; in such cases, we only release masked versions of the text together with annotation rather than the original wording, to comply with licensing requirements while preserving reproducibility. All models used are open-weight, allowing full release of data and code. Annotations were conducted by paid members of our laboratory, ensuring fair compensation. No sensitive or proprietary data were used, and the research complies with ethical guidelines for data collection and annotation.

## Acknowledgments

This work was supported in part by DARPA Grant No. D21AP10117 to Dr. Mark A. Finlayson. We thank the annotation team Amanda Chacin-Livinalli, Andrew Baad, and Ximena Puig for their contributions to the dataset development.

## 10. Bibliographical References


David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia.

AI@Meta. 2024. Llama 3 model card.

Béatrice Arnulphy, Vincent Claveau, Xavier Tannier, and Anne Vilnat. 2015. Supervised machine learning techniques to detect TimeML events in French and English. In *Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015)*, pages 19–32, Passau, Germany.

Mieke Bal and Christine Van Boheemen. 2017. *Narratology: Introduction to the Theory of Narrative*. University of Toronto Press, Toronto, Canada.

Roland Barthes. 1975. An introduction to the structural analysis of narrative. *New Literary History*, 6(2):237–272.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14, Atlanta, Georgia.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Seymour Benjamin Chatman and Paul A. Chilton. 1978. *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, Ithaca, New York.

Marian Roalfe Cox. 1893. *Cinderella: Three Hundred and Forty-five Variants of Cinderella, Catskin, and Cap o'Rushes*. David Nutt, London, UK.

Jonathan Culler. 2004. Story and discourse in the analysis of narrative. In Mieke Bal, editor, *Narrative Theory: Critical Concepts in Literary and Cultural Studies. Volume I: Major Issues in Narrative Theory*, pages 169–187. Routledge, London, UK and New York, USA.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth - github repository.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (NAACL-HLT 2019)*, pages 4171–4186, Minneapolis, Minnesota.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 671–683, Online.

Gérard Genette and Ann Levonas. 1976. Boundaries of narrative. *New Literary History*, 8(1):1–13.

Arthur C Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395.

Kristjan Greenewald, Luis Lastras, Thomas Parnell, Vraj Shah, Lucian Popa, Giulio Zizzo, Chulaka Gunasekara, Ambrish Rawat, and David Cox. 2025. Activated LoRA: Fine-tuned LLMs for intrinsics. *arXiv preprint arXiv:2504.12397*.

Jacob Grimm and Wilhelm Grimm. 1812/1988. *Kinder- und Hausmärchen*, volume 1. Realschulbuchhandlung, Berlin, Germany.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's English ACE 2005 system description. Technical Report 05-019, New York University, New York, New York.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. 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 2019)*, pages 434–444, Hong Kong, China.

Jerry R Hobbs. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Online.

David Hume. 1902. *Enquiry Concerning Human Understanding*. Clarendon Press, Oxford, UK.

Paul Jaccard. 1901. Etude de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547–579.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 254–262, Columbus, Ohio.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

David Lewis. 1973. Causation. *The Journal of Philosophy*, 70(17):556–567.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 73–82, Sofia, Bulgaria.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2023. Event extraction as question generation and answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 1666–1688, Toronto, Canada.

Mary McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3):276–282.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ines Montani and Matthew Honnibal. 2017. Prodigy: A modern and scriptable annotation tool for creating training data for machine learning models.

Lin Mu, Xiaoyu Wang, Li Ni, Yang Li, Zhize Wu, Peiquan Jin, and Yiwen Zhang. 2025. DenseLoRA: Dense low-rank adaptation of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, pages 10198–10211, Vienna, Austria.

Takaki Otake, Sho Yokoi, Naoya Inoue, Ryo Takahashi, Tatsuki Kuribayashi, and Kentaro Inui. 2020. Modeling event salience in narratives via Barthes' cardinal functions. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, pages 1784–1794, Barcelona, Spain (Online).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, Doha, Qatar.

Charles Perrault. 1697. Cendrillon, ou la petite pantoufle de verre. In *Histoires ou contes du temps passé, avec des moralités: Contes de ma mère l'Oye*. Claude Barbin, Paris, France.

Suhan Prabhu, Pranav Goel, Alok Debnath, and Manish Shrivastava. 2019. A language invariant neural method for TimeML event detection. In *Proceedings of International Conference on Natural Language Processing (ICON 2019)*, pages 36–44, Hyderabad, India.

Vladimir Propp. 1928/1968. *Morphology of the Folktale*, 2nd edition. University of Texas Press, Austin, Texas.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*, pages 28–34, Palo Alto, California.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 4512–4525, Online.

Shlomith Rimmon-Kenan. 1983. *Narrative Fiction: Contemporary Poetics*. Routledge, London, UK and New York, USA.

Gerard Salton. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Boston, Massachusetts.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A robust event recognizer for QA systems. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, pages 700–707, Vancouver, Canada.

Ge Shi, Yunyue Su, Yongliang Ma, and Ming Zhou. 2023. A hybrid detection and generation framework with separate encoders for event extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*, pages 3163–3180, Dubrovnik, Croatia.

Viktor Shklovsky. 1990. *Theory of Prose*. Dalkey Archive Press, Elmwood Park, Illinois.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Boris Tomashevsky. 1965. Thematics. In Lee T Lemon and Marion J Reis, editors, *Russian Formalist Criticism: Four Essays*, page 61–95. University of Nebraska Press, Lincoln, Nebraska.

Tom Trabasso and Linda L Sperry. 1985. Causal relatedness and importance of story events. *Journal of Memory and Language*, 24(5):595–611.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 2019)*, pages 5784–5789, Hong Kong, China.

David Wilmot and Frank Keller. 2021. Memory and knowledge augmented language models for inferring salience in long-form stories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 851–865, Online and Punta Cana, Dominican Republic.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, pages 38–45, Online.

Jeffrey M Zacks and Khena M Swallow. 2007. Event segmentation. *Current Directions in Psychological Science*, 16(2):80–84.

Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. 2025. LoRI: Reducing cross-task interference in multi-task low-rank adaptation. *arXiv preprint arXiv:2504.07448*.

Yu Zhong, Bo Shen, and Tao Wang. 2024. TGIN: Document-level event extraction with two-phase graph inference network. *Neural Networks*, 176(C):106343.

Mengna Zhu, Kaisheng Zeng, JibingWu JibingWu, Lihua Liu, Hongbin Huang, Lei Hou, and Juanzi Li. 2024. LC4EE: LLMs as good corrector for event extraction. In *Findings of the Association for Computational Linguistics (ACL 2024)*, pages 12028–12038, Bangkok, Thailand.

## 11. Language Resource References

Mark Finlayson, Jeffry Halverson, and Steven Corman. 2014. The N2 corpus: A semantically annotated collection of islamist extremist stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 896–902, Reykjavik, Iceland.

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.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML annotation guidelines version 1.2.1.

Marc Verhagen and James Pustejovsky. 2012. The TARSQI toolkit. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2043–2048, Istanbul, Turkey.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus.

# A. Prompts Used

## A.1. Kernel Prompt

```
SYSTEM_PROMPT = """Kernel events are
important events in a story that cannot
be deleted without destroying the logic
and they keep the story moving forward.
Each kernel event either initiates or
resolves an open possibility, advancing
the story towards the climax.  Kernel
events are causally linked:  each event
results from a preceding one and causes
those that follow.

Your task:  Identify kernel events in
the provided text.
- If there are kernel events, respond
with:  "The kernel event is:  [event]"
or "The kernel events are:  [event1],
[event2], ..."
- If there are no kernel events, re-
spond with:  "There are no kernel
events."
"""

INSTRUCTION_PROMPT = "Identify and list
all the kernel events described in the
following text."
```

## A.2. Event Prompt

```
SYSTEM_PROMPT = """TimeML events are
expressions that denote actions, pro-
cesses, states, or reporting as defined
in the TimeML framework.  They include
verbs or nominalizations that refer to
occurrences, situations, or states over
time.

Your task:  Identify TimeML events in
the provided text.
- If there are TimeML events, respond
with:  "The TimeML event is:  [event]"
or "The TimeML events are:  [event1],
[event2], ..."
- If there are no TimeML events, re-
spond with:  "There are no TimeML
events."
"""

INSTRUCTION_PROMPT = "Identify and list
all the TimeML events described in the
following sentence."
```