A Survey on Event-Based News Narrative Extraction

BRIAN FELIPE KEITH NORAMBUENA, Virginia Tech and Universidad Católica del Norte
TANUSHREE MITRA, University of Washington
CHRIS NORTH, Virginia Tech

Narratives are fundamental to our understanding of the world, providing us with a natural structure for knowledge representation over time. Computational narrative extraction is a subfield of artificial intelligence that makes heavy use of information retrieval and natural language processing techniques. Despite the importance of computational narrative extraction, relatively little scholarly work exists on synthesizing previous research and strategizing future research in the area. In particular, this article focuses on extracting news narratives from an event-centric perspective. Extracting narratives from news data has multiple applications in understanding the evolving information landscape. This survey presents an extensive study of research in the area of event-based news narrative extraction. In particular, we screened more than 900 articles, which yielded 54 relevant articles. These articles are synthesized and organized by representation model, extraction criteria, and evaluation approaches. Based on the reviewed studies, we identify recent trends, open challenges, and potential research lines.

CCS Concepts: • **Computing methodologies** → **Information extraction**; **Knowledge representation and reasoning**; • **General and reference** → **Surveys and overviews;**

Additional Key Words and Phrases: Computational narratives, narrative representation, narrative extraction, narrative analysis

ACM Reference format:
Brian Felipe Keith Norambuena, Tanushree Mitra, and Chris North. 2023. A Survey on Event-Based News Narrative Extraction. *ACM Comput. Surv.* 55, 14s, Article 300 (July 2023), 39 pages.
https://doi.org/10.1145/3584741

1 INTRODUCTION

Narratives are fundamental to our understanding of the world [1], and they provide a framework that enables humans to associate and represent events over time [15]. Moreover, narratives are a core element of collaborative sensemaking in society [8, 119]. In this context, narratives are defined as a *coherent system of interrelated stories* [42], where stories themselves are defined as

This work was supported by NSF grants CNS-1915755 and DMS-1830501, ANID/Doctorado Becas Chile/2019-72200105, and a Virginia Tech ICTAS Junior Faculty Award received by T. Mitra.
Authors' addresses: B. F. Keith Norambuena, Virginia Tech, Department of Computer Science, Torgersen Hall, Suite 3160, 620 Drillfield Dr. Blacksburg, VA, 24060, United States, and Universidad Católica del Norte, Department of Computing & Systems Engineering, Av. Angamos 0610, Antofagasta, Antofagasta, 1270709, Chile; email: briankeithn@vt.edu; T. Mitra, University of Washington, Information School, Mary Gates Hall (MGH), 1851 NE Grant Ln, Seattle SWA, 98195, United States; email: tmitra@uw.edu; C. North, Virginia Tech, Department of Computer Science, Torgersen Hall, Suite 3160, 620 Drillfield Dr. Blacksburg, VA, 24060, United State; email: north@vt.edu.

0360-0300/2023/07-ART300 $15.00
https://doi.org/10.1145/3584741

sequences of *events* [114]. These systems of stories help humans produce a shared understanding of the world [105]. In particular, extracting narratives from data is a fundamental task in our efforts to achieve this goal of common understanding [51].

In this survey, we focus on a specific type of narrative: news narratives. In particular, we analyze works that extract computational narrative representations from news articles. Work on general computational narratives started as early as the 1960s [94]. However, these early works focused mostly on narrative generation—usually through rule-based methods and grammars [3]—rather than extracting narratives from data. In contrast, the narrative extraction works reviewed in this survey start around the 2000s (e.g., [24, 76, 111]).

From an information retrieval standpoint, extracting narratives from data relies on several techniques from this field, including event [76] and entity extraction methods [14], as well as elements from search and ranking [61] and summarization techniques [55]. Furthermore, narrative extraction is supported by several artificial intelligence techniques, such as machine learning [106] and search and optimization [96].

Despite the importance of news narrative extraction, relatively little work has focused on clarifying the past trajectory and future agenda of news narrative extraction. Our goal with this survey is to fill this gap. This article presents a literature review of narrative extraction screening more than 900 papers from a variety of journals, conferences, and workshops. In particular, by thematically analyzing 54 articles, we identify a taxonomy of representations, extractions methods, and evaluation methods, which helps organize prior work and chart the path forward for future research. Taken together, all these elements provide a detailed account of the core elements of event-based news narrative extraction.

1.1 Scope of This Survey and Definitions

1.1.1 Narrative Definition. There are many potential definitions of *narrative* in the literature. General narrative theory focuses explicitly on understanding the general rules of narrative and its different arrangements that make it meaningful [1, 85]. The key intuition in formal narrative theory is that there is a distinction between the story itself and its representation. Narrative theory tries to understand the relationships between stories and their many possible representations [85]. Other definitions consider narratives as communication tools to construct a shared meaning of events with the purpose of influencing the behaviors [75].

Halverson et al. [42] define narratives not just as one story but rather as a *system of stories*. In other words, narratives are a systematic collection of interrelated stories with coherent themes. Stories are defined as sequences of *events* tied together in a coherent fashion. In this definition, events are the fundamental units of narrative action—they are either an act involving characters and entities or a happening where no entities are causally involved [1]. We leverage this definition to model news narratives. Thus, we have a series of hierarchical definitions starting from the *narrative*, then going into *stories*, and finally into the fundamental units of the narrative: the *event* and its related *entities*. Furthermore, these definitions require an underlying *order* for the events, as they have to be linked sequentially in the stories.

There are two fundamental units in our previous discussion: events and entities. These units provide different perspectives of the narrative—one is focused on the actions and happenings of the narrative, whereas the other is focused on the characters and other entities that participate in the events. However, to provide a more focused review, we will focus exclusively on *event-based narrative representations*. Thus, we define computational narrative representations as an *event structure* that represents different stories. We note that these event structures are discrete in nature (e.g., a graph or a timeline of events). Nevertheless, we note that some of the extraction methods that we review will leverage entity-based information, but they are not the focus of their representation.

Finally, we note that the simplest way to computationally represent a narrative is through a linear structure representing sequences of events (i.e., a timeline). In fact, this is the most common approach in our survey. However, we also find that there are more complex representations, based on event graph structures.

1.1.2 News Narrative Extraction. The main focus of this survey is on *narrative extraction* from *news data* ("How do we extract a news narrative from data?"). In particular, we focus exclusively on textual narratives extracted from a set of news articles published in traditional news sources—we exclude works that focus on mixed types of data (e.g., images and text, or videos and text). Thus, all of the surveyed works fall under the umbrella of natural language processing.

Moreover, we note that extraction can be performed at a document level (i.e., extracting a narrative from a single document) or at a corpus level (i.e., extracting a narrative from multiple news articles). As part of our scope definition, we focus on corpus-level extraction methods, where the goal is to obtain a narrative representation from a set of articles rather than on document-level extraction (e.g., extracting the narrative of a single document).

Throughout this work, we work under the assumption that most news articles focus on a single main event. This is a common assumption in story and narrative extraction methods [51] and a natural assumption when dealing with breaking news articles, as they are likely to present a single event [50]. We note that news articles may sometimes refer to previous or secondary events in their body, which can be used to link articles together. However, for the purposes of our definition, these references are not considered the *main event* of that news article. Following this assumption, we deal with three levels of resolution in our works: events as sentences, events as documents, and events as clusters. Events may be represented by relevant sentences extracted from a news article, and usually a single sentence is used for these purposes. Events may also be represented by an entire document (i.e., a news article). We note that there is some overlap between these two representations when documents are associated with headlines. Finally, events may also be represented as sets of documents that refer to the same main event.

We note that there are more granular views of events in the literature—for example, the notion of event from TimeML [74, 86], where events are a much more specific action (e.g., a perception or state) compared to a news event that may comprise multiple of these events [47]. Contrasting with the granular specifications of TimeML, there are also works that view events as sets of terms (e.g., keywords or entities) [103], akin to how topics are sometimes characterized in traditional topic modeling works [11], and construct timelines representing them as such. However, this view of events is too broad and lacks the specificity expected from news events. Thus, we do not consider narrative representations that use such approaches. Following these exclusion criteria, we removed approximately 10 articles from the final dataset.

Leveraging our previous discussion of narratives as a structured system of interrelated stories, we define the (event-based) narrative extraction task as follows:

News Narrative Extraction: Given a set of news articles, the news narrative extraction task generates a *discrete structure* comprised of *events* to represent the narrative.

We note that the structure is left deliberately ambiguous to allow for different types of representations, such as event timelines or event graphs. However, we note that all these overarching narrative representations are discrete in nature (e.g., event graphs), even if the underlying event representations could be continuous (e.g., text embeddings). Furthermore, the representation of the event itself can be defined in different ways depending on the *resolution* level (sentences, documents, or clusters) of the narrative representation. Furthermore, this definition excludes entity-based representations (e.g., character networks).

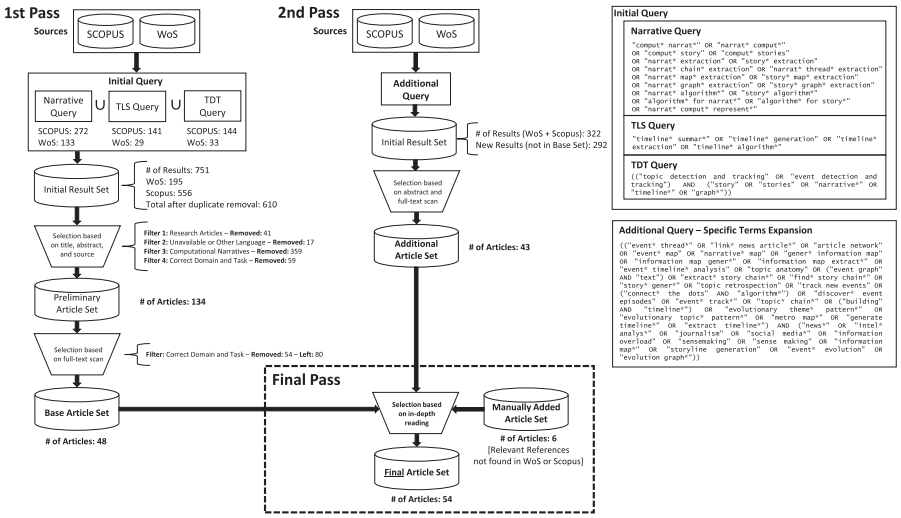

Fig. 1. Overview of the article collection process and the inclusion/exclusion criteria used to construct the final article set.

1.1.3 Exclusions: Related Tasks. We exclude works that focus on narrative generation, narrative forecasting, and narrative analysis. We also exclude works that only focus on representational issues without an associated method.

Narrative generation is a fundamentally different task from extraction that seeks to create new fictional narratives rather than extract a narrative that already exists (either fictional or non-fictional) [35, 36]. Furthermore, the focus of narrative generation is usually fictional narratives, not news narratives. Narrative forecasting (i.e., predicting the next events in the narrative) is a task that lies between extraction and generation, but its focus is on generating new events rather than on extracting the complete narrative [131]. Narrative analysis methods use existing extraction approaches to obtain a computational representation of the narrative and then use it to analyze the narrative [79, 95]. However, they do not provide new insight into the extraction task itself, unless they include a novel extraction method as well.

Moreover, we exclude interactive narratives, as these are a fundamentally different type of narratives where the story can be changed through user feedback and actions [19], which would not make sense in the context of news narratives. However, even though the underlying story cannot be changed, it might still be possible to interact with the narrative model. In fact, several works rely on interactivity at a presentation level.

Finally, we exclude works that focus on news narratives extracted from social media [10, 62], as social media narratives follow a different approach that requires not only analyzing content but also the users spreading it, leading to unique challenges that are left beyond the scope of this survey.

1.1.4 Inclusion and Exclusion Criteria. Having defined our scope, we provide details of our collection methodology and inclusion/exclusion criteria. We describe the query and steps used to generate the final article set in Figure 1.

We performed two article searches on SCOPUS and Web of Science. The first search was based on three queries that covered broad areas related to the news narrative extraction task: narrative extraction and computational narratives in general, **Topic Detection and Tracking (TDT)** [4], and **Timeline Summarization (TLS)** [37]. These latter two fields are highly related to our task

and provide a series of relevant works that we have examined in our survey. In particular, we note that most TDT works view news as flat collections [76] of events without an underlying narrative structure. Instead, we view news data as an interconnected *structure of events*. Nevertheless, some TDT works fit with our view of narratives and thus we include them in the review. In contrast, we consider most of the TLS line of works as a subset of the narrative extraction task and include many works from that field as part of the "event as sentences" resolution level. However, we exclude works that do not generate a full timeline and only focus on identifying relevant dates, as that is a different subtask. Next, we performed a second query based on a series of keywords obtained from the initial results. We applied the same inclusion and exclusion criteria for this second set of articles. After this, some additional articles that were not caught by our two main searches were added based on references from some reviewed articles. Finally, we performed a final pass on all the articles based on an in-depth reading of each article.

The rest of this article is structured as follows. The rest of Section 1 discusses related surveys and reviews. Section 2 presents an overview and summary of each one of the reviewed articles. Section 3 discusses the different extraction criteria. Section 4 presents a discussion of the evaluation approaches and metrics. Section 5 presents a discussion of our findings and future research directions. This survey concludes with a brief summary and key takeaways in Section 6.

1.2 Related Surveys

Most surveys regarding computational narratology focus on the task of narrative generation rather than extraction. In fact, there is an extensive series of survey papers and literature reviews on generation in conferences [112] and journals [36, 54] that cover narrative generation and its different approaches in depth. Moreover, there is even a book [69] on computational narrative representations for narrative generation and an extensive and in-depth book chapter on different cognitive approaches to narrative generation [80]. Narrative generation is also covered as a specific subtask of the more general field of natural language generation [35]. In contrast, general narrative extraction is not covered by any published survey. More specifically, our domain of interest—news narrative extraction—is also not covered in the literature. However, there are some surveys that touch on related topics. In the rest of this section, we provide a general description of these works and how they relate to our own survey.

First, we note a survey on the evaluation of summarization methods by Ermakova et al. [31] as a related approach to narrative extraction. In particular, this survey provides a comprehensive overview of existing metrics for the evaluation of narrative summarization methods. Narrative summarization is related to both narrative extraction and generation, as it requires extracting an internal narrative representation from data and then generating the summary. In comparison, our survey presents evaluation metrics for narrative extraction methods, some of which overlap with the evaluation metrics discussed in the aforementioned survey.

Second, we note the work of Richards et al. [91], which discusses representation models for narratives. Most of the discussion is specific to narrative generation, but there are general models that could be applied in both generation and extraction contexts. Nevertheless, the discussion is focused on what constitutes a narrative in general rather than being directly useful for the news narrative extraction task as defined here.

Third, we note the survey on extracting character networks from fictional narratives by Labatut and Bost [57]. Their work is related to ours as it focuses on the narrative extraction task, but with a much more specific scope focused on character-based models (i.e., entity-based narrative extraction). In contrast, our work has a different scope that considers event-based models. Moreover, their scope focuses on extracting networks from fictional narratives, whereas we consider extraction methods for non-fictional narratives in news data.

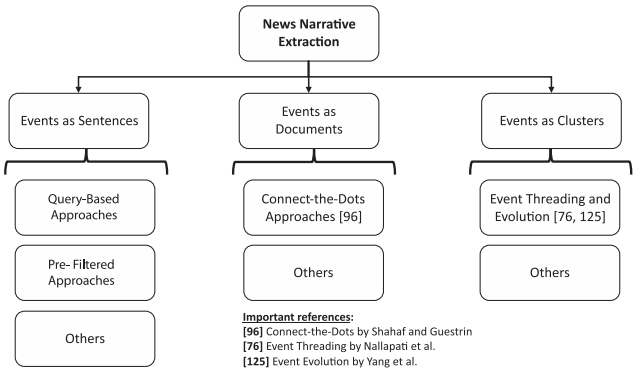

Fig. 2. Overview of the different methods used in news narrative extraction categorized by event resolution.

Finally, we note the recent survey on TSL approaches by Ghalandari and Ifrim [37]. Although there is plenty of overlap between this survey and our own, the news narrative extraction task that we cover is more general than just TLS, as we include methods that treat events as documents and clusters rather than at a sentence level. However, we highlight the empirical component of that survey, which includes an experimental section comparing the state-of-the-art methods in TLS.

2 NEWS NARRATIVE EXTRACTION

2.1 Overview

We found a total of 54 articles focusing on event-based news narrative extraction in our review. We present the articles based on the resolution level that they use: events as sentences, events as documents, and events as clusters. Figure 2 summarizes the identified approaches categorized by event resolution and some relevant subsets of these categories. In the sentence-level resolution, *query-based approaches* include an information retrieval step in addition to the narrative extraction itself. For example, these approaches require the user of the method to define a search query (e.g., "COVID" or "Terrorism") to find related documents in the dataset through similarity-based techniques or other methods before extracting the narrative from the queried subset. In contrast, *pre-filtered approaches* assume that the dataset has been already filtered and do not require an explicit query. In the document-level resolution, *Connect the Dots* approaches refer to the line of works derived from the seminal method of Shahaf and Guestrin [96] of the same name on storyline extraction. In the cluster-level resolution, event threading and evolution methods refer to a series of works based on the *event threading* concept of Nallapati et al. [76] or the *event evolution* concept of Yang et al. [125]. Works that fall under the "Others" do not fit in any of the defined subsets.

Table 1 summarizes the reviewed articles. In particular, we include the following columns in this table: *event resolution*, *number of stories*, *structure*, *type of approach*, and *event representation*. We now provide a brief description of these elements and their possible values.

Event resolution refers to the abstraction level at which the events are extracted. As mentioned in the scope definition, we consider three levels: sentences, documents, and clusters. Sentence-level works represent events as either a single sentence (e.g., the most important sentence or a headline) or a set of sentences (e.g., a sample of representative sentences). Document-level works represent events directly as a single document (e.g., a full news article). Cluster-level works represent events as sets of documents (e.g., multiple news articles that talk about the same basic event). Structure represents whether the extraction method generates a linear structure of events (e.g., a timeline [96, 123]) or a graph-like structure (e.g., a directed acyclic graph [51] or tree [67]). Figure 3 exemplifies these concepts.

Table 1. Summary of the Surveyed Articles

		Event Resolution			No. of Stories		Structure		Approach		Event Representation				
Year	Reference	Sentences	Documents	Clusters	Single	Multiple	Linear	Graph	Unsupervised	Supervised	Word Vectors	Topic Distribution	Neural Embeddings	Entities	Other
1998	Uramoto and Takeda [111]		×			×		×	×		×				
2004	Nallapati et al. [76]			×		×		×		×	×			×	
2004	Chieu and Lee [24]		×		×		×		×		×				
2005	Guha et al. [40]		×			×	×		×		×				
2006	Yang et al. [125]			×		×		×	×		×				
2006	Lin and Liang [64]			×	×			×	×		×				
2007	Lin et al. [63]			×	×			×	×		×				
2008	Chen and Chen [20]			×		×		×	×		×	×			
2008	Qiu et al. [87]			×		×		×		×	×				
2008	Lin and Liang [65]			×	×			×	×		×				
2009	Yang et al. [126]			×		×		×	×		×				
2010	Shahaf and Guestrin [96]		×		×		×		×		×				
2011	Yan et al. [124]	×			×		×			×	×				
2011	Yan et al. [123]	×			×		×			×	×				
2011	Hu et al. [46]	×			×		×		×		×	×			
2011	Khurdiya et al. [52]			×		×		×	×			×			
2012	Zhu and Oates [134]		×		×		×		×		×				
2012	Chen and Chen [21]			×		×		×	×		×	×			
2012	Shahaf and Guestrin [97]		×		×		×		×		×				
2012	Shahaf et al. [98]		×			×		×	×		×				
2013	Ansah et al. [5]	×			×		×			×	×				
2013	Li and Li [58]	×			×		×			×		×			
2013	Tran et al. [110]	×			×		×			×	×				
2013	Huang and Huang [49]	×			×		×		×			×			
2013	Tannier and Moriceau [106]		×			×		×		×	×				
2013	Shahaf et al. [99]		×			×		×	×		×				
2013	Shahaf et al. [101]			×		×		×	×		×				
2014	Nguyen et al. [77]	×			×		×		×		×				
2014	Zhu and Oates [135]		×		×		×		×		×				
2014	Huang et al. [48]			×		×		×	×		×	×		×	
2014	Wei et al. [116]			×		×		×	×		×				
2014	Hu et al. [47]		×			×		×	×			×			
2014	Zhou et al. [132]	×				×		×	×		×				
2015	Tran et al. [109]	×			×		×			×	×				
2015	Li et al. [60]	×			×		×			×		×			
2015	Bögel and Gertz [14]		×		×			×	×					×	
2015	Chen et al. [23]	×			×		×		×		×	×			×
2015	Shahaf et al. [100]			×		×		×	×		×				
2017	Wu et al. [121]	×			×		×		×		×				
2017	Liu et al. [67]			×		×		×		×	×				
2017	Laban and Hearst [56]		×			×	×		×		×				
2018	Wang et al. [115]	×			×		×			×		×		×	
2018	Tikhomirov and Dobrov [108]	×			×		×			×	×		×		
2018	Xu and Tang [122]			×		×		×	×				×		
2018	Zhou et al. [133]	×				×		×	×		×				
2019	Camacho Barranco et al. [18]		×			×		×	×		×	×		×	
2019	Cai et al. [17]			×		×		×	×		×				
2019	Yuan et al. [130]	×				×		×	×				×		
2020	Duan et al. [28]	×			×		×		×		×		×		
2020	Liu et al. [66]			×		×		×		×	×				
2021	La Quatra et al. [55]	×			×		×		×						×
2021	Yu et al. [129]	×				×		×		×			×		
2021	Liao et al. [61]	×			×		×		×				×		
2021	Keith Norambuena and Mitra [51]		×			×		×	×				×		

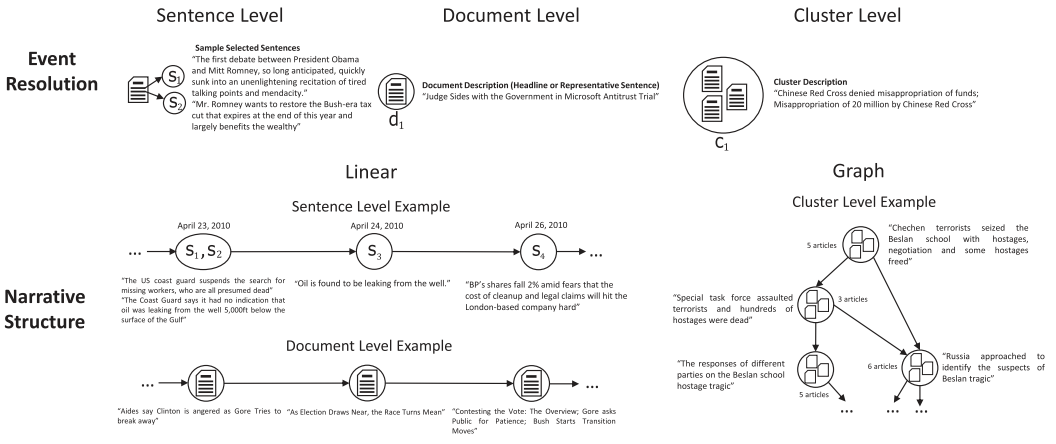

Fig. 3. Resolution level and narrative structure. Examples adapted from several works in this survey.

Number of stories refers to whether the method is designed to handle a single storyline or multiple storylines. Recall our definition of a story as a sequence of events. Most timeline extraction methods extract a single story, but some of them extract *parallel* timelines, where each timeline represents a different story from the data [56, 129]. In contrast, most graph-based works are designed to represent multiple storylines, due to their inherent more complex nature compared to

timelines. However, there are some works that represent a single story but provide extra information by exploiting graph structures—for example, appending additional nodes with related events to the central story [65].

Type of approach represents whether the method is supervised, which requires training data, or unsupervised, which does not require training data. In general, we considered any method where the authors had to train the model with labeled data before using it as supervised. However, some approaches only did this to find the optimal value of a small set of hyperparameters [60, 76, 124] and it could be possible to use them in an unsupervised manner, provided that those hyperparameters were fixed in some other way (e.g., heuristics or previous work information).

Finally, *event representation* provides information about the computational representation of the events. Note that this is separate from the resolution level of the event. In general, we found four types of representations: word frequency models (e.g., TF-IDF and bag-of-words vectors), topic distribution models (e.g., **Latent Dirichlet Allocation (LDA)** vectors), neural embeddings (e.g., BERT), and entity-based models (e.g., entity frequency vectors). Some works combine these approaches and have a mixed event representation that leverages all these elements in some way to extract the final narrative model. There are some works that did not fit in any of these approaches and were marked as "Other."

2.2 Events as Sentences

We start with works that use a sentence-level resolution. Most of these methods fall under the umbrella of TLS [37]. However, not all of them fit with traditional TLS work. We split the discussion into three parts: query-based approaches, pre-filtered approaches, and others.

2.2.1 Query-Based Approaches. These approaches perform an information retrieval step before or during the narrative extraction process based on a user-defined query. In some cases, the query just acts as a simple filter, and in others, they explicitly include the query into the narrative extraction model.

Chieu and Lee [24] present a query-based timeline extraction approach where each event is represented as a sentence. This is the earliest form of the "events as sentences" that we could find in the literature. Sentences are first filtered based on the query and then ranked according to two criteria: *interest*, based on the frequency of the reported event in the query, and *burstiness*, based on the idea that important events form clusters around their date of occurrence. To determine whether two sentences are reporting the same event, the authors use cosine similarity. Furthermore, interest is determined based on a time window to avoid combining events that should be separated due to their temporal distance. To reduce *redundancy*, duplicated sentences are removed based on a time window around an important event that depends on the interest value.

Yan et al. [124] proposed a TLS method based on balanced optimization and iterative substitution of sentences. Their optimization problem is defined in terms of *relevance*, *coverage*, *coherence*, and *diversity*. All these terms are based on the **Kullback-Leibler Divergence (KLD)** [53] of the summary items with a target distribution. Relevance is related to a user-defined query and is defined as the KLD between the summary items and the internal representation of the query. Coverage is based on a global term–KLD between the summary items and the whole corpus–and a local term–KLD between the summary items and the set of sentences from the same date. Coherence is defined locally, based on the KLD between each summary item and its neighboring summaries by using an exponential temporal decay term (i.e., consecutive dates should have relatively similar summaries). Diversity is measured across dates and measures the average KLD of each sentence with respect to all other sentences in a leave-one-out manner. The final utility function is a weighted average of these terms with user-defined weights and can be defined at a local level (to evaluate individual

time periods) and a global level (to evaluate the full timeline). To find the sentences, this utility function is optimized in an iterative manner by replacing sentences in the date summaries and improving the utility value in each step using a dynamic programming algorithm that considers both local and global constraints.

Li and Li [58] propose a topic modeling approach for timeline extraction from news called the Evolutionary Hierarchical Dirichlet Process (EHDP) to capture the evolution pattern of news topics. This model extends Hierarchical Dirichlet Process models [107] by incorporating time dependencies and background information. In particular, it adds a new dynamic Dirichlet mixture model. Using this proposed topic model, a series of sentences are selected to represent each time period in the timeline based on the weighted average of three criteria: *relevance* (the summary should be related to the overall query), *coverage* (the summary should generalize the important topics in each time period), and *coherence* (each summary should be coherent with neighboring time periods). To score these criteria, the authors propose a topic scoring algorithm based on KLD that leverages their new topic model. The selected sentences are used to represent the relevant events in each time period.

RaRE (Rank and RErank) [77] is a system for building timelines of events from news articles based on a user query. In particular, it extracts timelines in three steps: temporal clustering based on salient dates, event relevance and salience scoring, and sentence re-ranking using an iterative algorithm that seeks to reduce redundancy. The method has an underlying assumption that each document represents a single event that can be described by a single sentence. The temporal clustering step identifies salient dates based on the *number of occurrences* of the date in the documents. The sets of events linked to a specific salient date are called *temporal clusters*. Furthermore, as a preprocessing step, events are clustered into *thematic clusters* inside each date using hierarchical clustering based on normalized Manhattan distance and a user-specified threshold. The event relevance and salience scoring steps use these criteria to rank events (i.e., documents) inside each temporal cluster. In particular, it uses four metrics: *event relevance*, *thematic cluster relevance*, *event salience*, and *date salience*. Event relevance is based on cosine similarity with the initial query. Thematic cluster relevance is based on the similarity of its thematic cluster with the initial query based on the average relevance of each event in the cluster. Event salience is based on the frequency of terms on a specific date. Date salience is based on the (normalized) total relevance of all events happening on that date. Finally, the sentence re-ranking step measures the frequency of unused terms on each date for a specific event to reduce redundancy.

Another topic modeling approach uses a time-dependent Hierarchical Dirichlet Tree model [60] to capture the evolution of news topics using the Dirichlet Tree distribution—a generalization of the Dirichlet distribution [26]. In particular, the model represents topic distributions in sentences using a tree of fixed depth. Each sentence is associated with a path and with a topic vector and each node has its own topic distribution over words. Using the proposed topic model, sentences are selected by first locating candidate words on the nodes of the tree based on the **Jensen-Shannon (JS)** divergence of sentences and KLD between word collections. Next, the candidate sentences are scored based on the weighted average of the following criteria: *focus* (the timeline should be relevant to a given query), *coherence* (the sentences should be correlated), and *coverage* (the sentences and documents should be representative).

Wu et al. [121] propose a sentence-based approach to generate timelines. In particular, all the sentences that contain a user-defined query word are split by date and used to generate a *date vector* representing that specific date. Sentences that do not include parseable dates are grouped based on *similarity* with the date vector. All sentences are then ranked based on similarity with their corresponding date vector and unrelated sentences are filtered out based on a user-defined threshold. The highest-ranking sentence is used to summarize each date.

Tikhomirov and Dobrov [108] propose a news timeline generation approach from a query based on three steps: query extension, inter-document graph extraction, and intra-document sentence ranking. Query extension is based on *pseudo-relevance feedback* and consists of three query levels, which are constructed using the most significant terms based on TF-IDF weights. Next, as a preprocessing step, dates that have a *frequency* below a statistically determined threshold are discarded. The next two steps use an *inverted pyramid* [50] heuristic, which assumes that the upper part of the article contains the most important information and the lower part of the article may contain *references* to important events from the past. In particular, the inter-document graph extraction step constructs a *similarity matrix* between the upper and lower parts of the documents. If the similarity is above a specified threshold, then the articles are considered to be linked, creating a similarity graph. Next, a ranking algorithm—LexRank [30]—is used to determine the importance of each document. Documents that are above a specified importance threshold are used to further expand the original query one more time. Finally, to rank the final selected sentences for the summary, a ranking metric is defined by taking into account content similarity (using cosine similarity) with the extended query (i.e., maximizing relevance) and subtracting similarity with already extracted sentences (i.e., minimizing redundancy).

WILSON (neWs tImeLine SummarizatiON) [61] is a query-based TLS method for news based on a divide-and-conquer approach consisting of two major components: date selection and text summarization for each selected date. For date selection, the method first tags temporal expressions in sentences and constructs a date reference graph based on these annotations. Next, the method assigns weights to the edges of the date reference graph by taking the product of the *number of references* and *temporal distances* with the references. Then, it uses the PageRank algorithm [82] on the extracted graph to find the most salient dates. However, this approach leads to a bias toward older dates, as they have had more time to get references. Thus, the model is augmented with an exponential *recency adjustment* weight, which is used to initialize the Personalized PageRank algorithm [9], which allows for non-uniform initial distributions. Next, the daily summarization can be done using any multi-document summarization approach. Specifically, the authors use TextRank [73] based on BERT [27] representations to generate the summaries.

2.2.2 Pre-Filtered Approaches. These approaches assume that the dataset has already been filtered as part of a preprocessing step. Thus, they do not explicitly model the query in their extraction model.

Yan et al. [123] propose a system to generate news timelines using a trans-temporal summarization approach, where the summary for each time period depends on its context—that is, nearby time periods. Before generating the timeline, the system chooses the important time periods (e.g., specific days) to be summarized based on *burstiness*. The timeline extraction approach is based on two components: a *global component*, which defines the structure of the overall summary and the inter-temporal relationships between each period of the timeline, and a *local component*, which defines the summary in each time period. The global component is based on a global graph that uses inter-date dependency, which is computed using *temporal proximity* and a global *affinity* model for each sentence based on PageRank. Furthermore, to ensure a diverse set of sentences in the global component, the system incorporates DivRank into the affinity model [70] to penalize the lack of *diversity* in the sentence selection. Next, the local component is based on a local sentence graph for each time period following a similar approach to the global graph. To generate the final sentence selection in each time period, the system optimizes a weighted ranking generated by both components.

Hu et al. [46] propose a timeline overview method for news based on the concept of breakpoints—points in time where a significant development or change occurs (i.e., important

events). Their extraction approach consists of three steps. First, they analyze *topic activity* using a Topic-Activeness Hidden Markov Model and discard inactive periods. In practice, this is done by measuring whether there is new information using KLD and document frequency. Next, the breakpoints are identified by detecting topic variations in each time period using a topic mixture model, in particular, a generative probabilistic mixture model [71], and a Theme-Transition Hidden Markov Model to model topic evolution. Specifically, breakpoints are identified by using JS divergence to measure *topic variation* between two consecutive time points. Then, a summary for each breakpoint is generated by selected representative sentences—based on Jaccard *similarity with topic keywords* and *relevant entities*.

Tran et al. [5] present a supervised learning method to extract timelines from news articles based on linear regression. Their model first identifies salient dates based on *burstiness* (i.e., high-frequency periods), then selects the most representative sentences from the news articles on each of these dates. In particular, the model uses *surface-level features* (e.g., length and position of the sentences), *coherence features* (e.g., causal and temporal signals), *topic features* (e.g., TF-IDF information and cross entropy), and *time-related features* (e.g., popularity over time and use of temporal expressions) to determine the key sentences of each date. Subsequent work by Tran et al. [110] used SVM-Rank instead of linear regression and expanded upon this supervised framework. In particular, they leverage three metrics to evaluate the event sentences: *relevance*, *novelty*, and *continuity*. Relevance is learned using the SVM-Rank mentioned earlier. Novelty is evaluated by measuring the non-overlapping *n*-grams over the total *n*-grams between a candidate sentence and previously selected sentences. Continuity is a measure of local coherence—there should be smooth transitions in the timeline—that is computed as the average *n*-gram overlap of all sentences in the current day with the previous summary. The final score is based on a weighted average of these metrics. To learn the relevance function, the authors leverage the same set of features from their previous work [5], but they also added an extra set of features about the event itself. For example, they evaluated whether the sentences properly represent the main event of the article, using the fact that the first sentences should contain the most relevant information (following the *inverted pyramid structure*). Thus, they evaluate the similarity between the sentence summary and the first four sentences. Once the SVM-Rank method was trained, the ranking was fed to a dynamic programming approach to optimize the final score.

Huang and Huang [49] present an event storyline generation method based on a mixture-event-aspect probabilistic model that can detect and distinguish the different types of subevents in the article dataset. Their model is an extension of Probabilistic Latent Semantic Analysis [44] and LDA [12]. In particular, their model detects *global* aspects (i.e., terms that are important throughout the whole story) and *local* aspects (i.e., terms that are important in a specific event inside the story). Based on the extracted aspect model, the *bursty* periods for each aspect are extracted to measure their popularity on a certain date and detect relevant events. Based on these results, it is possible to extract a timeline and select the most representative sentences associated with both global and local aspects to compose the final storyline with adjustable weights for the aspects. Sentences are selected by minimizing the overall *information loss* over each aspect. In particular, the LexPageRank algorithm [29] is used to rank sentences and KLD is used for sentence similarity.

Tran et al. [109] propose a TLS approach based on article headlines. Their approach is based on a random walk model using a topic-sensitive version of PageRank [43] that selects relevant headlines from the dataset for each time period. There are three key metrics to evaluate the relevance of a headline: *informing value*, *spread*, and *influence*. The informing value depends on whether the headline provides factual information or an opinion, review, or another non-informing category. It is a binary value computed using a supervised learning approach based on an SVM classifier to separate facts from opinion [128]. Influence tries to measure the impact of an event on future events

(e.g., "the president resigns" would lead to a "new election" event) based on references from future events using similarity between future news articles and the headline of the event. Spread is based on the intuitive idea that a relevant event will be reported in multiple news outlets—that is, its reporting will be spread over multiple headlines. Thus, it is a measure of *positive* redundancy and is formally defined as the probability of a headline being duplicated. To estimate whether two headlines are duplicates, the system uses a supervised logistic regression model trained on semantic similarity measures based on paraphrase detection literature [72]. Having defined these elements, the goal is to maximize all three aspects to select the best headlines. This is done by using PageRank on a graph of headlines, taking into account both spread (graph edges) and influence (random walk probability), to generate the final rankings. Next, to generate the final timeline for each day, the resulting rankings are selected greedily, subject to redundancy constraints, informativeness constraints, and a maximum number of headlines per day.

Chen et al. [23] present a supervised TLS algorithm based on aging theory for news datasets. Aging theory [22] is a model that tracks the life cycle of events using an energy function, which increases when an event becomes popular and diminishes with time. The method works by extracting sentences (i.e., specific events) and the publication time from news articles and using a classification model built with SVM to determine whether they belong in the output timeline. This approach is based on *surface-level features* (e.g., noun frequencies and stop word frequencies), *importance features* (e.g., latent semantic analysis scores), *topic features* (e.g., topic word frequencies), an *aging score feature* (i.e., changing coverage of an event over time), and a *novelty feature*. The aging score is used to measure the life cycle of each term over time using a recurrence relation with TF-IDF representations. The novelty score is based on the Jaccard similarity of the current summary and the candidate sentence.

2.2.3 Others. Here we present works that use a sentence-level resolution but differ from the majority of the other works that follow the traditional TLS approach. In particular, we consider works on extracting disaster storylines from news and works that present variations on the traditional TLS task.

Disaster Storylines. The works of Zhou et al. [132, 133] present a framework to construct spatio-temporal storylines for disaster management from news data based on how the disaster location moves over time (e.g., a typhoon moving through different areas). This approach generates timelines for two levels of representation: a *global level* that follows the progress of the disaster through each location and a *local level* that focuses on a specific location. To extract the storyline, a series of snippets (i.e., event sentences) are extracted from the news articles using named entity recognition methods and grouped together based on a similarity graph. Then, a set of representative sentences is selected by finding the minimum dominating set [102] using a greedy algorithm. Next, an integer linear programming approach is used to select the optimal sequence for the main route of the disaster by maximizing the *coherence* of the story chain, subject to a series of *structural*, *chronological*, and *length* constraints. In this method, coherence is defined based on consecutive content similarity rather than word influence. However, the key difference is that this formulation includes a *smoothness* constraint, which is specifically designed to track the moving location of disasters through time. Smoothness is based on simulating the natural trajectory of a disaster. In particular, the constraints set a maximum distance for consecutive events (i.e., avoiding jumps to locations too far away) and seek to avoid acute angles that could be formed by two consecutive connections (i.e., avoiding sharp turns in the trajectory of the disaster). Once the main storyline has been constructed, the next step is to analyze the local level storylines. For each main storyline event, a set of similar articles is selected and used to construct a multi-view graph that represents the event relationships based on content similarity. Then, a Steiner tree algorithm is used on the multi-view graph to generate a local storyline for that location.

Yuan et al. [130] propose dTexSL, a disaster storyline extraction approach that extends the works of Zhou et al. [132, 133]. Unlike the previous approach, the news articles are first divided into different subsets based on location and are represented using neural embeddings. Locations are found by measuring the distance of the locations described in each article—using named entity recognition to find location references—and merging locations that are close enough based on a user-defined threshold. Then, an integer linear programming approach is used to select the *key locations* (i.e., document clusters). Instead of choosing events to maximize coherence like before, the goal is to maximize the *number of documents covered* on the map. The model has similar constraints as the original approach: *chronological order*, *length*, and *smoothness*. Once the main storyline has been constructed, a word embedding method is used to construct a multi-view graph that represents the event relationships based on content similarity. Using this graph, a set of representative articles are selected based on two criteria: *uniqueness*, computed using information gain, and *relevance*, computed using a measure of node importance. Then, a dynamic Steiner tree algorithm is used on the multi-view graph to generate a local storyline for that specific location. Finally, a traditional multi-document summarization method [39] is applied to generate a high-level event description for that specific location.

Task Variations. Duan et al. [28] introduce another variation on the TLS task called *comparative TLS*. In this task, the goal is to provide timelines consisting of major contrasting events from two datasets. Their approach is based on three core characteristics: *coverage*, *distinctness*, and *diversity*. Coverage is based on the idea that the timelines should cover most of the important information or topics from each dataset. Distinctness is based on the idea that the events in a timeline should be distinct from the events on the other timeline at each time point, to allow for a proper contrast between them. Diversity is based on the idea that each timeline should cover a diverse set of events from its dataset. To model these attributes, the authors propose a dynamic Markov model that is built around sentence similarity at a document level for each timestep. In particular, sentences are selected from news articles to describe events based on local and global importance measures through the use of an affinity-preserving mutually reinforced Markov random walk model based on the PageRank algorithm. The output is a timeline that contains contrasting events from both datasets.

Yu et al. [129] propose a variation on the basic TLS task, called **Multi-Timeline Summarization (MTLS)**. In this task, events are represented as sets of sentences and computationally represented by the neural embedding model sentence-BERT [90]. Given a set of timestamped news articles, MTLS seeks to automatically extract timelines for important and different stories found in the dataset. The authors propose a framework to solve this task called *2SAPS* (Two-Stage Affinity Propagation Summarization). There are two key components in their framework: an event generation module and a timeline generation module. The event generation module seeks to extract important events from the document collection. To do so, it uses an *affinity* propagation approach to cluster similar sentences [34] and to identify the event of the article and any other previously *referenced* event. Furthermore, there is a *temporal similarity term* that uses an exponential decay function to penalize similarities of events that are temporally far away. Once the events are identified, a subset of these events is selected based on a weighted average of a , based on event frequency, and a *consistency metric*, based on the intra-event similarity. Next, the timeline generation module has three internal steps: event link, time selection, and TLS itself. Event linking is based on the weighted average between a *co-reference score* (based on entities or terms shared between events) and *semantic similarity* (e.g., cosine similarity). Based on these average scores, the system builds an event graph and uses affinity propagation on it to determine the initial clusters (i.e., timeline sets). Next, there is a timeline selection based on the weighted average of , the average event salience of the timeline, and *timeline coherence*, the average semantic similarity scores between chronologically adjacent

events. The timeline summarizing step selects an exemplar sentence for each event in the timelines as the most typical and representative member of each event. Finally, there is an add-on timeline tagging step that assigns a label to each timeline, based on the most frequent words of the events.

Summarize Dates First [55] is a TLS pipeline that follows a different paradigm for TLS based on generating a summary for each individual date first, then selecting the most relevant dates using these summaries. This is different from the traditional approach where the relevant dates are selected first. Furthermore, this approach aggregates dates by leveraging higher-level temporal references (i.e., references to previous events in the article). Summarize Dates First consists of three steps: temporal tagging, per-date summary extraction, and summary-drive date selection. In the temporal tagging stage, the raw text is annotated to identify date-level references (e.g., 31 December 2021) and high-level references (e.g., last December). The per-date summary extraction step uses any traditional sentence-based summarization algorithm from the multi-document summarization literature (e.g., TextRank [73]). Summary-driven date selection is the last step and uses a selection strategy, called *Graph-Based Date Selection*, which uses graph ranking algorithms (e.g., PageRank, HITS). In particular, a directed date graph model is built using the temporal references of the dataset, where the edge weight connecting two dates is influenced by the count of date-level references and the similarity between the date summary and the high-level references to the earlier date.

2.3 Events as Documents

Here we present works that use a document-level resolution. We split the discussion into two parts: methods that build upon the Connect the Dots approach by Shahaf and Guestrin [96]—a seminal work in the field of news narrative extraction—and others. We further divide the presentation based on whether the methods are linear based or graph based. We note that the works cataloged as others did not have a discernible pattern beyond using a document-level resolution.

2.3.1 *Connect the Dots Approaches.*

Linear Representations. Shahaf and Guestrin [96] proposed the Connect the Dots algorithm to extract temporal chains of documents (i.e., timelines). In particular, they use an optimization approach that seeks to maximize the overall *coherence* of the timeline. Coherence measures the smoothness of a storyline, and a coherent story should not have drastic changes in content or topic. To implement this metric, they propose an approach based on , a measure of word relevance computed through random walks on a word-document graph, and *word activations*, which measure whether a specific word is active at a given point in the storyline. To extract the story chains, they used linear programming to maximize coherence subject to structural and temporal constraints. However, since linear programming provides non-integer solutions, it required additional heuristics to find the best chain by defining a rounding method. The linear programming approach used in the original Connect the Dots implementation was computationally expensive. Thus, Shahaf and Guestrin [97] proposed a new method to reduce computational costs and avoid the approximate solutions from the linear program. In particular, they used a best-first search algorithm based on an extension heuristic—given a chain of documents, adding a new document to the chain will at most keep the same level of coherence—and the original linear program to individually evaluate each chain.

Expanding upon the Connect the Dots method, Zhu and Oates [134] proposed an algorithm to extract story chains from newswire articles that connect two user-defined endpoints based on the following characteristics: *relevance* (the articles on the chain should be relevant to the endpoints), *coherence* (the transition between events should be smooth), low *redundancy* (there should only be one representative article for every event of the chain), and *coverage* (the chain should cover every important event). To compute measures of these characteristics, the article proposes using random

walks on bipartite graphs formed by articles and words, where the weights are given by TF-IDF representations. Thus, based on these criteria, their proposed algorithm consists of two iterative stages. The first phase consists of a divide-and-conquer bisecting search problem that adds articles to the story chain. In particular, in this phase the algorithm finds the best article to insert in the middle of each current link of the story chain (i.e., it bisects the current links) based on coherence and relevance criteria. The second phase consists of pruning redundant articles, by removing a certain percentage based on how much coherence they would add to the current story chain, and irrelevant articles, by removing events that are similar to each other and temporally close with an exponential decay function. These phases are repeated until there are no more articles to add or prune. A subsequent article by the same authors [135] revisits the story chain algorithm and extends this approach by adding an intermediate clustering step that groups documents into document clusters and words into word clusters. These clusters are used to generate a new bipartite *correlation graph* that combines the weight of individual documents and words through a weighted average to assign the edge weights. Furthermore, the model adds a named *entity bias* that assigns a higher weight to named entities compared to other terms. This is modeled through a co-occurrence frequency matrix for entity pairs, which is then used to compute a relevance score for each document in the dataset based on the named entities. In turn, these elements are used to modify the cluster and document weights in the correlation graph.

Camacho Barranco et al. [18] propose a storyline extraction algorithm that takes a set of user-defined articles as a seed and generates a timeline of articles based on a series of evaluation metrics. First, the authors propose a temporal criterion to filter candidate documents based on a range between the latest publication date of the seed articles and a maximum threshold away from the earliest publication date of the seed articles (i.e., in the interval $[t_{\min} - t_{threshold}, t_{\max}]$). Next, there is a topical criterion that measures how much a candidate article can deviate from the seed articles based on KLD and LDA topics. Having defined their basic framework, the authors then formalize an optimization problem to extract the storylines by selecting article connections based on different criteria: *incoherence*, *similarity*, *overlap*, and *uniformity*. Incoherence is based on the average pairwise Soergel distance between documents—measured using TF-IDF information for the entities of the document—with a temporal factor to penalize temporally distant articles. Similarity is used as a penalty factor to enforce diversity in non-adjacent articles of the storyline, implemented as a negative exponential factor based on the Soergel distance. Both of these metrics are weighted by a relevance factor of the documents and are smoothed using modified Gaussian distributions to measure event overlap. Next, an overall overlap factor for the storyline is computed, assigning a penalty based on the difference between publication dates and a user-defined threshold. The overlap factor ensures that the breakpoints occur at sufficiently distinct dates. The uniformity penalty seeks to avoid the case where the optimal solution selects purely irrelevant events as optimal by penalizing uniform weights. The objective function to minimize consists of the sum of the product between incoherence and similarity, multiplied by the overlap and uniformity penalties.

Graph-Based Representations. Metro maps [98, 99] are an extension of the Connect the Dots approach that represents more than a single storyline using a directed acyclic graph of events. In particular, the metro maps method is a structured summarization approach that captures the evolution of multiple stories and their interactions. The stories are represented using a metro map metaphor, where each metro line represents a story and stations represent key events. Metro lines intersect in specific stations, representing how storylines connect with each other. This representation is extracted by solving an optimization problem. In particular, the goal is to maximize *connectivity*, subject to *coverage* and *coherence* constraints. Coverage is computed based on how well specific terms or keywords are represented in the selected events and is defined using a *submodular function* that encourages diversity (e.g., if a term is already covered, adding a document that covers

it provides little extra coverage). These keywords depend on the specific corpus or domain of application. Coherence is defined following previous work of Shahaf and Guestrin [96, 97]. Finally, connectivity is defined as the number of stories that intersect, which is used to ensure that the final metro map is connected. The optimization problem is solved in phases. First, a series of coherent candidate metro lines are selected based on a divide-and-conquer approach, which constructs long lines from shorter ones and encodes them in a graph. Then, the method extracts a set of coherent lines that maximize coverage using an approximation algorithm based on the submodularity of the coverage function (otherwise finding these lines is an NP-hard problem). Finally, connectivity is increased using a local search approach that substitutes lines without sacrificing coverage.

Similar to the metro maps metaphor, the narrative maps model [51] provides a framework to extract and represent narratives based on a route map metaphor. The narrative and its stories are shown as a series of routes through landmarks, which represent the events. In computational terms, the narrative is modeled through a directed acyclic graph of events. The events are represented through neural embeddings of article headlines. The graph is extracted by solving an optimization problem defined following a linear programming formulation similar to the Connect the Dots approach. The optimization problem is based on maximizing *coherence* subject to *coverage* constraints. Coherence measures how much sense it makes to connect two events together and is defined as the geometric mean of the *content similarity* of events, using cosine or angular similarity, and their *topical similarity*, based on JS similarity of their topic distributions based on clustering. Coverage is measured by the average percentage of topical clusters covered by the selected events based on their topic distributions. Once the optimal map has been found, the main storyline is extracted by normalizing the coherence values of the edges into probabilities and finding the maximum likelihood path. Then, a set of representative landmarks (i.e., important events) of each story by finding the maximum antichain, which corresponds to the point of the maximum width of the graph.

2.3.2 Others.

Linear Representations. Guha et al. [40] propose an *event threading* approach based on a graph decomposition method that generates document timelines. In particular, they propose decomposing a directed acyclic graph into disjointed node paths that ensure that as many nodes as possible participate in at least one path (i.e., they seek to maximize a notion of *coverage*). The first step is to construct the graph, and they propose doing this based on *important terms* (or even entities) in the document collection and their co-occurrence. Furthermore, documents are modeled following a bag-of-words approach, although the method is also designed to handle TF-IDF representations. Once the graph is constructed, the next step is to solve the event thread extraction problem. To do this, they propose three formulations: an exact algorithm, a maximum approach, and a dynamic programming approach. The first method is an exact algorithm based on minimum cost flow, which has a high computational cost and is impractical. The second is an approximation algorithm based on maximum matching in bipartite graphs that solves the thread extraction problem for a fixed maximum size. The third method is based on an approximation algorithm that uses dynamic programming to solve the thread extraction problem for a range of thread sizes.

Laban and Hearst [56] present newsLens, a system to build and visualize long-ranging news stories. In particular, their system groups news articles based on their , based on a graph clustering approach, and then selects a sample of headlines from salient dates, based on the *frequency* of publications. In more detail, the first step in their extraction approach is to construct a keyword graph for a starting time period using TF-IDF representations of the articles. Next, a local topic graph is created based on a user-defined threshold for the number of shared keywords between articles. After the initial time period, a sliding window approach with a user-defined length is used to handle the rest of the data. For each time period, a local topic graph is created and compared with

the graph from the previous period to check for three types of relationships: linking (connecting a topic from the current graph to a pre-existing topic), splitting (dividing a pre-existing topic into new topics in the current period), or merging (combining separate topics from the previous step into a single one of the current period). However, this approach is not able to handle stories that have *long-time gaps* between publications. To handle these cases, the content similarity of non-overlapping stories is analyzed and merged if above a specific threshold. Afterward, their method assigns a name to the storyline by extracting noun phrases from the news articles and scoring them based on multiple criteria (e.g., length, type of noun, abstractness, and frequency). Finally, salient dates are selected based on local frequency changes, and representative headlines are sampled randomly from these dates to generate the final timeline visualization.

Graph Representations. Uramoto and Takeda [111] proposed a graph-based approach to model the relationships between news articles. In particular, they use a directed graph based on temporal ordering and *event similarity*. This is the earliest article that fits with our definitions of event-based narrative representations for news narratives that we found. In particular, the authors use the concepts of *genus* and *differentia* words. For adjacent articles, genus words are computed using the intersection of their word sets and represent already known information in the story. In contrast, differentia words are built from the set difference between the articles (in temporal order) and represent new knowledge in the story. Thus, differentia words are more important when trying to find coherent sequences of articles. The events are represented with a variation of TF-IDF that assigns more weight to *differentia* words.

Tannier and Moriceau [106] propose an approach for building multi-document event threads from news articles. In particular, they use a supervised learning approach with a series of classifiers to define the type of relationship between news articles: *same-event*, *continuation*, or *reaction*. The output of this method is a temporal event graph, where the nodes correspond to events (represented as news articles) and the edges are labeled with the corresponding relationships. In particular, the first step is to determine whether there is a connection at all between the articles. To do so, an initial classifier is implemented using a series of *content similarity* features (e.g., word overlap, cosine similarity, and similarity of the first sentences) to construct the initial temporal graph. However, this is not enough to find all potential relationships and a second-level classifier is included that takes into account the results from the previous classifier by using *degree-based features* from the temporal graph. Next, after a connection has been established, another classifier determines whether this connection is based on the articles referring to the same news event, same-event connection, or based on a continuation, when an event is a direct continuation or consequence of a previous one. This classifier relies on *date-based features* (e.g., differences in publication time, date references, and references between events themselves) and *keyword-based features* (e.g., usage of temporal words, reaction words, or opinion words). The output is fed into another classifier that leverages degree-based features again to find more relationships. Due to the *transitive* nature of the same-event and continuation relationships, a post-processing step takes the graph and constructs the *transitive closure* for these specific relations. Afterward, a final classifier uses the same features to determine whether a continuation is a reaction—a subset of continuations that relate the reactions of people (or organizations) to an event.

Hu et al. [47] propose a system to model storyline interactions from news events. Their approach generates a series of event timelines focusing on specific entities or topics and their interactions with each other. In particular, this results in a directed graph connecting multiple events. In contrast to other approaches, the underlying representation of events is based on the *main event descriptors* (i.e., the answers to Who, What, When, Where, Why, and How) [50], which are extracted directly from each article and represent the key elements of the event. Based on this information, a coherence graph is constructed and used to identify the storylines through a

random walk. Coherence is defined by three factors: *subtopic consistency*, *entity relatedness*, and *time continuity*. To measure subtopic consistency, the first step is to use a generative probabilistic mixture model to discover latent subtopics. Then, JS divergence is used to measure the distance of topic distributions between articles. Next, entity relatedness is measured by the average *affinity* of the entities from each pair of articles using normalized point-wise mutual information. The time continuity factor is simply defined as an exponential penalty term dependent on the temporal distance between events. The coherence graph is built by creating edges between documents that have a coherence score above a given threshold. Based on the coherence graph, a series of *informative events* that connect multiple storylines are identified. Specifically, a topic-sensitive PageRank algorithm [43] is used to discover these events. In turn, these events feed the storyline generation algorithm, an iterative algorithm that selects a single informative event for each story for each day.

Bögel and Gertz [14] present a temporal linking framework based on the concept of article references. In particular, they exploit the structure of news articles to construct an information network. Instead of comparing articles based on overall content similarity, they exploit the use of *lead paragraphs*, *explanatory paragraphs*, and *additional information paragraphs* in typical news articles. Specifically, they construct the network based on *temporal expressions*, *keywords*, and *entity names*. To select valid event connections, the first step is to filter based on temporal information contained in the text based on a temporal tagger. Next, connections are evaluated based on the *similarity* of the lead paragraph of a news article with all the other paragraphs of another news article (i.e., capturing references to the event). Similarity is computed based on the entities and keywords mentioned in each paragraph based on a weighted average of Jaccard and cosine similarity. Finally, irrelevant edges are pruned based on a user-defined threshold. However, some non-relevant edges are kept if they fulfill the role of a *support path*—paths that have non-relevant edges but share endpoints with fully relevant paths—that provide more evidence of two events being connected. The output is a directed graph based on references, not necessarily acyclic, as there are future temporal references in some articles.

2.4 Events as Clusters: Event Evolution and Threading

Now, we present works that use a cluster-level resolution. We divide the discussion into two parts: works related to event threading [76] and evolution [125], and others.

2.4.1 Event Threading and Evolution. Nallapati et al. [76] use a directed graph model to represent to capture the structure and dependencies of events in a news topic. They call this extraction process *event threading*. They represent each event as a cluster of news articles. Event threading is a supervised method that consists of two phases: clustering documents and modeling dependencies. The clustering process starts with a cluster for each document in the dataset and merges them iteratively based on similarity until the similarities fall below a predefined threshold. The authors evaluate three types of cluster similarity on the average link, complete link, or single link of the clusters based on document similarities. Document similarities are based on *content similarity* (e.g., cosine similarity), *common locations*, and *common entities*. Furthermore, there is an exponential decay term based on the *temporal distance* to penalize larger temporal distances between documents. Next, dependency modeling uses surface-level features of the document clusters, such as word distributions and time ordering of the news articles. Based on this information, the authors propose several link extraction criteria (complete link, simple threshold, nearest parent, best similarity, and maximum spanning tree). These approaches rely on temporal order, similarity information, or structural information.

SToRe (Storyline-based Topic Retrospection) is a topic retrospective system [63–65] that extracts the main storyline from a given news topic and provides a summary of the topic based on

this storyline. In particular, the extraction process consists of four phases: event identification, topic structure identification, main storyline construction, and storyline-based summarization. In the event identification phase, similar news articles will be clustered together to represent a single event using self-organizing maps. In the topic structure identification step, the events are linked together based on whether their *similarity* exceeds a specific threshold. To compute similarity, the events are represented with a vector of term weights using the concepts of *genus* and *differentia* words [111]. Then, cosine similarity is used to compare the event vectors. Next, in the main storyline construction step, an MST is extracted from the constructed topic structure. The MST is based on the relevance of each event with respect to the topic. The MST is used to generate a timeline of events, and it is further extended with small side branches of other relevant events based on a specific threshold. Finally, in the storyline-based summarization, a summary is generated for each event based on the news articles contained in its cluster using accumulated weight summary [39].

Yang et al. [125, 126] use directed acyclic graphs to represent the evolution of events in online news. They call their approach *event evolution graphs*, which represent temporal and causal relationships between events. Events are defined as sets of news articles and are represented as the average of the TF-IDF vectors of each article they contain. We note that the proposed method assumes that events and their corresponding articles are already computed. In practice, this would require a clustering step before constructing the graph. These events are linked together based on their similarity and a user-specified threshold, which is computed based on *content similarity* (e.g., cosine similarity), *temporal proximity*, and *document distributional proximity* (which penalizes bursty periods with many articles about the same event). The latter two terms are represented through exponential decay factors. Furthermore, users are able to reduce the temporal granularity of the event evolution graph, which merges specific events that occur in short time frames.

Qiu et al. [87] propose another event evolution graph extraction method. Their construction method follows an iterative approach based on *content similarity* and *temporal order*. In particular, documents are first grouped into clusters using the OHC method [88] in the first time period, which gives rise to the initial events. Next, the PRAC method [89] is used to build classifiers and determine whether the documents of the next time period are continuations of a cluster identified in the previous period. If so, a new event node is created using the identified cluster as its parent. This process is repeated until the last time period. Next, *twigs*—paths that die before the end of the timeline—are removed based on a user-set tolerance, and equivalent event nodes are merged to reduce graph complexity.

TSCAN (Topic Summarization and Content ANatomy) [20, 21] is a method to analyze news data that produces a global summary and constructs an event evolution graph. We focus on the event graph component of this method. First, news articles are grouped into themes obtained through a matrix factorization approach with TF-IDF document representations. Next, the news articles of each theme are temporally segmented using an *energy value* threshold based on eigenvalues from the matrix representation. In practice, this generates clusters of documents based on *frequency*, which are associated with the nodes of the event evolution graph. The evolution graph is a directed acyclic graph, where the edges are constructed using *temporal similarity*, computed using the temporal distance between events with special cases to consider event overlap, and *content similarity*, based on cosine similarity.

Khurdiya et al. [52] propose a system that extracts directed graphs to represent stories from news data using multi-perspective links. Each node of this graph is associated with multiple news articles. The system uses LDA to extract topics in each time unit (e.g., a day). The extracted topics are associated with sets of articles based on the strength of the topic in each article and form the basis of the story identification model. We note that these topics and their article sets correspond

to the notion of event that we use in this survey. Next, article sets are linked chronologically based on *topic correlation* (e.g., Pearson's correlation coefficient) and a user-defined threshold, generating a directed graph of events.

Wei et al. [116] identify event episodes in news datasets and construct a temporal episode graph (i.e., an event graph under our definitions in the survey). In particular, this article shows a discovery mechanism that organizes news documents into events using novel TF-IDF representations that incorporate a temporal component. Then, the system builds a link structure based on inter-cluster similarity measures. The first proposed event representation, called $TF\text{-}IDF_{\text{Tempo}}$, gives more weight to features with *consecutive occurrences* in a sequence of documents (i.e., it incorporates the surrounding context of the document) by modifying the IDF component of TF-IDF to consider the order of the documents. However, this approach is too strict and is unable to model overlapping events. Moreover, it also has a high bias toward low-frequency articles that are temporally close. Thus, the authors propose a second representation, called $TF\text{-}Enhanced\text{-}IDF_{\text{Tempo}}$, which modifies the IDF component by adopting the *significance factor* proposed by Luhn [68] and a *temporal gap threshold* to allow for short discontinuities in feature appearances. These representations are used with Hierarchical Agglomerative Clustering (HAC) [113] to construct the article clusters that represent the events. For the purposes of clustering, document similarity is defined by content similarity (e.g., cosine similarity) and a negative exponential penalty for temporally distant documents.

Huang et al. [48] propose a different event evolution approach to build and analyze event relationships based on three types of event connections. In particular, they define a *co-occurrence dependence relationship*, an *event reference relationship*, and a *temporal proximity relationship*. The authors define events as a set of news articles and identify them through clustering and topic modeling using a combined similarity measure that leverages LDA and a TF-IDF document model with cosine similarity. Once the events are identified, the method extracts a series of core features (i.e., key entities and terms of the article) by analyzing the *lead* of the articles and evaluating whether their frequency is above a specified threshold. These core features are used to construct a vectorial representation of the events. For the co-occurrence relationship, the method computes the aggregation of all mutual information between all features of the event, generating a symmetric matrix that represents all event-event relationships. For the event reference analysis, the method identifies shared core features and defines the degree of event reference based on the frequency of references in an event to the core features of a previous event, adjusted by the weight of these terms in the referencing event. Temporal dependency is evaluated using an exponential decay formula.

Event Phase Oriented News Summarization (EPONS) [115] is a TLS approach that assumes that a story summary contains multiple timelines, each one corresponding to a specific event. To model the semantic relations of news articles, EPONS uses a graph model, called the *Temporal Content Coherence Graph*, which is an event graph based on two metrics: *content coherence* and *temporal influence*. Content coherence is based on the weighted average of *topic level similarity*, modeled by JS divergence over an LDA topic distribution, and *entity-level similarity*, modeled over a ranking of named entities using the Tanimoto coefficient. Temporal influence is modeled through a Hamming (cosine) kernel to properly separate temporally distinct events. The Temporal Content Coherence Graph is built by selecting edges that are above user-specified thresholds in each metric. Based on this graph, EPONS uses a modified structural clustering approach to group the news articles into different events. Furthermore, small clusters of similar articles are filtered out to ensure that the events are modeled properly. This post-processing is done by using four quality metrics on a pretrained logistic regression classifier: percentage of new articles, time interval length, pairwise topic similarity, and pairwise entity similarity. Having identified the events, it is now necessary to construct the individual summaries and finalize the timeline. To do

so, a vertex-reinforced random walk [70, 84] is used to rank the relevance of news articles inside each event, in a similar manner to PageRank. Next, a supervised model is used to determine whether the headlines are factual (i.e., they are reporting a specific event) or an opinion, as opinion-based headlines are not considered useful for timelines and must be filtered out. Finally, an optimization method is used to maximize the total *relevance*, subject to *non-redundancy* constraints (i.e., disallowing events that are too similar) to select the news articles.

Cai et al. [17] propose a method to extract **Temporal Event Maps (TEMs)** based on the *content dependence* degree and *component event reference* degree for each pair of events. TEMs are directed graphs that have events as nodes, relations as edges, edge weights representing the strength of event relationships, and node weights representing the importance of each event. Events are defined as groups of related documents and identified using an LDA model. After obtaining the events, the next step is to compute the two core metrics that define the TEMs. The content dependence degree is defined as the aggregation of all mutual information among the features of each event. The content reference degree is defined by the presence of *core features* of an event—salient terms based on frequency—in other events. Unlike content dependence, this is not a symmetric relationship between events. To construct the TEMs, the first step is to order events based on starting time. Then, connections are added for events that surpass a user-specified threshold for the product of content dependence and event reference degrees, which provides the edge weights for the graph. Finally, a ranking procedure based on PageRank is used to generate the event importance values.

2.4.2 Others.

Information Cartography. Continuing with their work on metro maps, Shahaf et al. [100, 101] propose a new framework called *information cartography* that features *zoomable* metro maps, allowing users of the map to visualize the news at different levels of resolution, and allowing the user to zoom in to specific metro stops and generate a new map. Metro stops and events are no longer represented as single documents but as clusters of events. The articles are segmented into time windows and clusters are computed using a community-detection algorithm on word co-occurrence graphs. To extract the maps, an optimization problem is defined based on finding the best *structure* for the map, relying on the idea of minimizing the total number of storylines (to reduce unneeded complexity) and maximizing the *number of covered clusters* (to ensure that the stories are well covered). This approach leads to simple stories being modeled as a single metro line and more complex stories requiring the use of multiple shorter lines. Furthermore, a series of additional constraints for *story coherence*, *cluster quality*, and *map size* is imposed.

Building upon the concept of metro maps and information cartography, Xu and Tang [122] propose a narrative representation in the context of societal risk events (e.g., earthquakes) called *risk maps*. These maps follow the same basic representation of information cartography with events being represented as clusters of documents. However, one key difference is that this approach leverages advances in text representation by using neural word embeddings for news articles before clustering. To obtain the risk map, the authors choose to maximize *coverage* as their primary objective, followed by *connectivity*, subject to a minimal *coherence* constraint. Coverage is defined based on how well each cluster is covered by the different storylines. Connectivity is simply the number of storylines that intersect. Coherence is defined based on the Jaccard similarity of consecutive clusters in the storylines. The optimization problem is solved using a greedy algorithm that finds the best path among clusters at each step.

Story Forests. Liu et al. [66, 67] propose the Story Forest approach, where different stories are constructed and represented as a forest of event trees. First, events are clustered using a community detection approach on word co-occurrence graphs using betweenness centrality. Next, documents are associated with each topic through a similarity based on TF-IDF representations. Afterward,

a second step groups documents together based on a supervised classifier (SVM) to determine whether pairs of documents refer to the same event based on TF-IDF features and similarities between the contents and titles of articles. Story Forest is built iteratively by adding events into its trees by using three operations: merge, extend, and insert. Before adding the events, it is necessary to determine the correct story tree. This is done based on a measure of *compatibility*, computed as the Jaccard similarity of the keywords of the event and the tree. If no trees are related to the event, a new tree is created with the event as its root. To add the event to an existing tree, the method first tries to *merge* it with any of the existing events into the same node using the previously trained SVM classifier. Otherwise, the method scans all the nodes to identify which tree to *extend* based on a measure of connection strength determined by three elements: *compatibility*, *coherence*, and *time penalty*. Compatibility is measured by the similarity of their centroids based on cosine similarity. Coherence is a story-level measure that takes into account the path of events from the root of the tree to the newly appended event by measuring the average consecutive compatibility value. Finally, the time penalty is an exponential decay factor that depends on temporal distance. If none of the events are appropriate, the event is *inserted* as a new node connected to the root.

3 NARRATIVE EXTRACTION CRITERIA

In this section, we present a summary of the different construction criteria found in the reviewed articles. These criteria refer to either an evaluation metric or additional information used in the extraction algorithms themselves as part of an objective function (e.g., coherence optimization), selection criteria (e.g., filtering based on content similarity or topic distribution similarity), and other types of extraction heuristics (e.g., leveraging article structure to compute content similarity or evaluating the use of opinionated language). The first part of Table 2 provides an overview of the different construction criteria. We note that these criteria are not mutually exclusive and can be combined as needed.

Relevance. Relevance metrics evaluate whether the events in the narrative are relevant or significant to a given query or topic [58, 60, 77, 108, 123, 124]. In general, relevance is measured by borrowing techniques from traditional search methods in information retrieval, such as PageRank and its variations [109, 115, 134, 135]. However, some approaches use supervised methods to learn a ranking function [5, 110]. The results from such techniques are used to feed other parts of the algorithm or could be directly used to select relevant events, turning this issue into more of a traditional information retrieval problem rather than a narratological one.

Content Similarity. Another approach to extracting narratives is based on modeling content similarity between events. More than two-thirds of the methods use some sort of content similarity measure. There are many ways to do this—in particular, we found the following approaches: surface-level similarity comparisons (e.g., Jaccard similarity or cosine similarity) [46, 77, 111], topic similarity based on topic distribution information (e.g., comparing topic vectors extracted from LDA models) [23, 60], and entity-based comparisons (e.g., entity co-occurrence in events) [76, 135].

The exact choice of approach is highly dependent on the event representation. In recent years, researchers have started leveraging advances in text representation with neural embeddings (e.g., BERT) [28, 51, 61, 108, 129], which have several advantages over traditional frequency-based models and are better able to capture semantic similarities.

The use of entity-based information in event-based narrative extraction methods to measure event content similarity remains limited in scope, with sparse usage over the years compared to other content similarity measures [14, 18, 46, 47, 76, 115, 135]. Combining entity information with other types of similarities would provide a much more holistic view of content similarity. Furthermore, expanding upon this approach, content similarity metrics could exploit the *main event descriptors* [50] to compute a more precise similarity measure.

Table 2. Summary of the Extraction Criteria and Evaluation Metrics Used in the Reviewed Articles

		Extraction Criteria														Evaluation Metrics									
Year	Reference	Relevance	Surface Similarity	Topic Distribution	Entities	Coherence	Coverage	Dispersion	Diversity and Redundancy	Output Structure	Article Structure	Content References	Temporal References	Burstiness and Frequency	Temporal Distance	Traditional IR Metrics	Summarization Metrics	Ranking Metrics	Clustering Metrics	Coherence	Coverage	Dispersion	Diversity and Redundancy	User Performance	User Perception	Other or N/A
1998	Uramoto and Takeda [111]		×																							×
2004	Nallapati et al. [76]		×		×										×	×										
2004	Chieu and Lee [24]		×											×	×										×	
2005	Guha et al. [40]		×				×																			×
2006	Yang et al. [125]		×											×	×	×										
2006	Lin and Liang [64]		×																					×	×	
2007	Lin et al. [63]		×													×										
2008	Chen and Chen [20]		×											×	×		×									
2008	Qiu et al. [87]		×																							×
2008	Lin and Liang [65]		×																					×	×	
2009	Yang et al. [126]		×											×	×	×										
2010	Shahaf and Guestrin [96]					×																			×	
2011	Yan et al. [124]	×				×	×		×								×									
2011	Yan et al. [123]	×												×	×		×									
2011	Hu et al. [46]		×	×	×											×										
2011	Khurdiya et al. [52]			×												×										
2012	Zhu and Oates [134]	×	×			×			×																×	
2012	Chen and Chen [21]		×											×	×		×									
2012	Shahaf and Guestrin [97]					×																			×	
2012	Shahaf et al. [98]					×	×			×														×		
2013	Ansah et al. [5]	×	×	×		×					×		×	×		×	×									
2013	Li and Li [58]	×		×		×	×										×									
2013	Tran et al. [110]	×	×	×		×					×		×	×			×									
2013	Huang and Huang [49]			×										×			×									
2013	Tannier and Moriceau [106]		×							×			×			×										
2013	Shahaf et al. [99]					×	×			×														×		
2013	Shahaf et al. [101]					×	×			×														×		
2014	Nguyen et al. [77]	×	×	×					×					×		×	×									
2014	Zhu and Oates [135]	×	×		×	×			×																×	
2014	Huang et al. [48]		×	×								×			×	×										
2014	Wei et al. [116]		×											×	×	×										
2014	Hu et al. [47]			×	×										×	×	×								×	
2014	Zhou et al. [132]		×			×				×							×									
2015	Tran et al. [109]	×							×				×	×		×									×	
2015	Li et al. [60]	×	×	×		×	×										×									
2015	Bögel and Gertz [14]		×		×						×		×			×					×					
2015	Chen et al. [23]		×	×											×	×	×									
2015	Shahaf et al. [100]					×	×			×														×		
2017	Wu et al. [121]		×																							×
2017	Liu et al. [67]		×			×									×				×						×	
2017	Laban and Hearst [56]			×										×	×											×
2018	Wang et al. [115]	×		×	×				×						×	×		×								
2018	Tikhomirov and Dobrov [108]	×	×								×	×		×			×									
2018	Xu and Tang [122]					×	×			×											×	×				
2018	Zhou et al. [133]		×			×				×							×									
2019	Camacho Barranco et al. [18]		×	×	×	×			×													×			×	
2019	Cai et al. [17]		×									×				×		×								
2019	Yuan et al. [130]						×			×							×									
2020	Duan et al. [28]						×		×							×	×						×			
2020	Liu et al. [66]		×			×									×				×						×	
2021	La Quatra et al. [55]		×										×				×									
2021	Yu et al. [129]		×			×						×		×	×		×									
2021	Liao et al. [61]												×				×	×							×	
2021	Keith Norambuena and Mitra [51]		×	×		×	×																		×	

Coherence. Coherence metrics evaluate whether the narrative makes sense. Due to their importance as an extraction metric, we show some mathematical formulations of coherence and coherence-like metrics in Table 3.

Although coherence has a formal definition in narratological terms [1], it is just as complex and ill defined as relevance in computational terms. One particular motivation for the definition of

Table 3. Sample of Different Formulations of Coherence from the Reviewed Articles

Formula	Resolution Level	Description	Source
$\frac{1}{\lvert S\rvert - 1} \sum_{(c_i, c_j) \in S} \text{JaccardSim}(c_i, c_j)$	Events as Clusters	This is a measure of coherence based on average Jaccard similarity along a story S based on cluster words.	[122]
$\frac{1}{\lvert S\rvert - 1} \text{CosineSim}(c_i, c_j)$	Events as Clusters	This is a measure of coherence based on average cosine similarity along a story S based on cluster centroids.	[67]
$\max_{\text{activations}} \left\{ \min_{(d_i, d_j) \in S} \sum_{w \in \text{Words}} \text{Influence}(d_i, d_j \lvert w) \cdot \mathbb{1}(w \text{ active in } d_i, d_j) \right\}$	Events as Documents	This is the full form of the coherence for a storyline S from the original Connect the Dots algorithm. It is based on maximizing the sum of word influences over active words in the storyline. Influence can be changed for any other type of scoring mechanism.	[96]
$\min_{(d_i, d_j) \in S} \text{CosineSim}(d_i, d_j)$	Events as Documents	This is a measure of coherence based on the minimum cosine similarity along a story S based on document vectors.	[132]
$\min_{(d_i, d_j) \in N} \sqrt{\text{SurfaceSim}(d_i, d_j) \cdot \text{TopicSim}(d_i, d_j)}$	Events as Documents	This is a measure of coherence for a narrative N based on the minimum geometric mean of surface-level similarity (e.g., cosine similarity) and topic-level similarity (e.g., JS divergence). It is based on document vectors and topic distribution vectors..	[51]
$\frac{\sum_{i,j \in \lvert D\rvert \times \lvert D\rvert} w_i \cdot w_j \cdot \Phi \cdot \text{Soergel}(d_i, d_j) \cdot \lvert t_i - t_j\rvert}{\sum_{i,j \in \lvert D\rvert \times \lvert D\rvert} w_i \cdot w_j \cdot \Phi}$	Events as Documents	This is a measure of incoherence rather than coherence. It is based on the average Soergel distance and includes a temporal distance term as well. The events are weighted by their relevance (w_i and w_j) and their temporal distance using a custom kernel Φ.	
$\sum_{s \in E} \frac{\text{Count}_{\text{match}(s,E)}(\text{gram}_n)}{\text{Count}_{\text{Previous}(E)}(\text{gram}_n)}$	Events as Sentences	This is a measure of coherence based on the n-gram overlap between the current event sentences and the sentences of the previous summary of the timeline.	[18, 110]
$\frac{1}{1 + \exp\left(JS(E, \text{Previous}(E))\right))}$	Events as Sentences	This is a measure of coherence based on the JS divergence between the current event sentence and the previous event sentence of the timeline.	[60]
$\frac{\sum_{\delta=-\Delta/2}^{\delta=\Delta/2} \exp\left(-\delta/\upsilon\right) \cdot KLD(E_t, E_{t-\delta})}{\sum_{\delta=-\Delta/2}^{\delta=\Delta/2} \exp\left(-\delta/\upsilon\right)}$	Events as Sentences	This is a measure of coherence based on KLD between the current event at time t and all the other local events in a Δ time window surrounding the event. The events are weighted by their temporal distance based on parameter υ.	[58]

coherence that stands out is the idea of *smoothness* from the Connect the Dots [96, 97, 100, 101] series of works. In particular, they use the concept of *word influence* and *word activations* (i.e., the sustained importance of the word in a storyline) to construct stories that have smooth transitions.

Other approaches compute coherence based on content similarity. These works also seek to generate *smooth* stories by avoiding drastic local changes based on content similarity [18, 51, 60, 66, 67, 110, 122, 124, 132–135], without explicitly defining active words or topics like the original Connect the Dots approach. Finally, one approach also considers coherence around the idea of causality [5, 110] in a supervised setting (e.g., causal signals in text).

Coverage-Like Metrics. Coverage-like metrics evaluate whether the extracted narrative properly covers the relevant events, stories, or topics. These metrics include coverage itself and related metrics, such as redundancy and diversity. The most basic form of coverage is simply the percentage of topics or relevant events covered by the extracted representation (or some variation of this metric) [28, 40, 51, 130], or a probability estimation [98–101, 122]. Equation (1) shows an example formulation of coverage for a cluster c, where Π represents an extracted narrative with storylines l.

$$\text{Cover}_{\Pi}(c) = 1 - \prod_{l \in \Pi} (1 - \text{Cover}_{l}(c)) \tag{1}$$

Another approach to compute coverage is to do a content similarity comparison between the output and the full dataset (or a relevant subset) [58, 60, 124]. In contrast, redundancy and diversity [28, 115, 124, 134, 135] metrics are based on the idea that events should not be covered more than necessary, thus high redundancy can lead to coverage problems.

Structural Information. Some works evaluate the structure of the output narrative representation. In particular, these metrics consider aspects such as size (in general) or connectivity (in graph-based narratives).

Size can be used as a proxy for complexity (e.g., length of the timeline) [122]. In most cases, rather than as an evaluation metric, size is used as a constraint (e.g., setting a maximum story length) [98–101, 130, 132, 133].

Connectivity metrics [98–101] are used to ensure that narrative graphs avoid isolated stories, as they should be interwoven throughout the narrative. Structure metrics are mostly analyzed at a global level (e.g., the total number of connected stories). However, it is possible to consider local structural features, such as node degrees [106].

Exploiting the internal article structure [14, 108, 110] is another piece of structural information used by some methods. Most breaking news articles are written following the *inverted pyramid structure* [50], where the most important information—the main event descriptors—is shown first in the *lead*. Thus, the first few lines of an article describe its main event [110], and subsequent paragraphs may contain more details and reference previous events.

Content References. Another criterion to consider in news narrative extraction is the use of content references. As mentioned before, some news articles make explicit references to previous works in their body. Note that this differs from explicit date-based references discussed previously, which rely on explicit temporal information. This approach also differs from general content similarity because of its goal of identifying specific references rather than global similarity.

One way to identify these references is to compare the lead of a news article with the additional information paragraphs of another article [108]. Other approaches identify references based on sentence co-occurrence without considering article structure [129]. Alternatively, a set of core features [17, 48] (e.g., relevant keywords or main event descriptors) could be identified and used to detect references in other articles. Once identified, these references can be used to identify relevant events based on reference-based metrics (e.g., bibliographic coupling).

Temporal Features. Temporal information, such as the temporal distance between events or specific date references, has been used. In particular, temporal distance is commonly used to penalize events that would otherwise be similar in content. For example, consider two articles describing separate protests in a city, one during the year 2000 and another in the year 2010. These two articles would likely be quite similar in terms of content, including both surface-level features and topic distributions. However, given the temporal separation between them, they would likely refer to different events. Thus, a common strategy is to define an exponentially decreasing term of the form $C_0 \exp\left(\frac{-\Delta t}{\sigma}\right)$ (or similar), where C_0 and σ are predefined constants [47, 48, 66, 67, 76, 116, 123, 125, 126, 129], although there are other approaches, such as kernels to perform temporal proximity projections [115, 123] or overlap-based measures [20, 21]. However, we note that the use of a temporal penalty is not always desired. Some events are continuations of stories that did not have anything new to report for a long time. For example, the investigation results of a flight accident might come much after the accident itself has been covered, leading to temporal gaps in story coverage [56, 116]. Thus, it is necessary to distinguish between continuations and completely new storylines when the time gap is high enough.

Burstiness and frequency measures and metrics based on these (e.g., energy values) are other time-based criteria used to identify relevant events and dates [5, 20, 21, 23, 49, 56, 77, 108, 109, 116, 123, 126, 129]. For example, periods with many publications are likely to contain important events. Alternatively, a specific event might be reported several times by different outlets. Finally, other temporal features include the use of specific temporal expressions or date references in the text [5, 14, 55, 61, 106, 109, 110] to identify temporal cross-references between documents.

4 EVALUATION METRICS

In this section, we discuss the evaluation approaches for the narrative output of the extraction methods. In particular, we show the evaluation metrics used to assess the quality of the extracted

narratives. These output metrics are generally intended to be interpreted by humans, unlike the extraction criteria which may or may not be easy to interpret. In particular, user-based evaluation metrics (e.g., task performance or user perception) are an important subset of output evaluation criteria. The second part of Table 2 provides an overview of the different output evaluation criteria.

4.1 Computational Metrics

These metrics seek to evaluate the extracted narrative based on computational measures of narrative quality. These metrics are usually supervised, requiring a gold standard dataset to be computed.

4.1.1 Supervised. We first discuss supervised approaches. In particular, we identified three broad types of metrics here: traditional information retrieval metrics, summarization metrics, and ranking metrics.

Traditional Information Retrieval Metrics. Several works—about a third of the reviewed articles—rely on classical evaluation metrics such as accuracy, precision, recall, and the F_1 score [14, 17, 23, 28, 46–48, 52, 63, 76, 106, 109, 115, 116, 125, 126] taken from traditional information retrieval and machine learning literature. In particular, these approaches evaluate the quality of the output by measuring whether *events* or their *connections* were identified correctly. Some methods also use variations of these basic metrics, such as the mean average precision [5, 77] over multiple dates.

Summarization Metrics. Specialized metrics from the summarization domain have also been used to evaluate narratives in several works—about a third of the reviewed works use them.

In particular, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics [81] have been used to evaluate the output of narrative extraction methods, mostly in TLS works with a sentence-level event resolution [5, 23, 49, 55, 58, 60, 61, 77, 108, 110, 124, 129], but also in works with a document-level resolution [47, 130, 132, 133]. ROUGE metrics include variations such as ROUGE-N (which measures the overlap of N-grams), ROUGE-L (which measures the longest common subsequence), ROUGE-W (a weighted version of ROUGE-L that favors consecutive subsequences), ROUGE-SU (skip-bigram and unigram-based co-occurrence statistics), and their precision, recall, and F_1 score variants. The most common variant is ROUGE-N, which we show in Equation (2).

$$\text{ROUGE-N} = \frac{\sum_{s \in R} \sum_{\text{gram}_n \in s} \text{Count}_{\text{match}}(gram_n)}{\sum_{s \in R} \sum_{\text{gram}_n \in s} \text{Count}(\text{gram}_n)} \tag{2}$$

In Equation (2), n represents the length of the n-gram, and R represents the reference summaries (i.e., the ground truth). $\text{Count}_{\text{match}}(\text{gram}_n)$ represents the maximum number of n-grams that co-occur in a candidate summary and the reference summaries. $\text{Count}_R(\text{gram}_n)$ represents the number of n-grams in the reference summaries.

An alternative is to measure the average *summary-to-document content similarity* where the summary is compared against the documents in the dataset using a text similarity measure (e.g., cosine similarity) [20, 21].

We note that these metrics are mostly used with linear representations rather than graph-based models—only three of the reviewed works that extract graphs use summarization metrics [20, 21, 47]. This contrasts heavily with the case of traditional information retrieval metrics, where the split was much more balanced between linear (~40%) and graph representations (~60%). This might be due to the inability of these metrics to handle complex structures.

Ranking Metrics. Other works rely on ranking-based metrics, like those used in traditional search tasks from information retrieval. For example, Wang et al. [115] use a relevance-based approach to evaluate their event phase summaries. Liao et al. [61] evaluate the ranking performance of WILSON with the mean reciprocal rank and discounted cumulative gain [7]. Cai et al. [17] use the normalized discounted cumulative gain [127] to evaluate all their events.

Clustering Metrics. Liu et al. [66, 67] used clustering metrics to evaluate the event nodes—which are represented as clusters of articles—in their Story Forest method. In particular, they use the homogeneity, completeness, and V-measure scores [93]. These metrics require labeled datasets to be computed, thus they are supervised despite being designed to evaluate unsupervised clustering methods. In particular, homogeneity is larger when each extracted cluster only contains members of a single class. In contrast, completeness is maximized when all members of a true class are in the same cluster. Finally, the V-measure takes both of these metrics and computes the harmonic mean between them, similar to how the F_1 score treats precision and recall in traditional classification metrics. We note that none of the other events as clusters methods used these metrics or other similar clustering metrics to evaluate their models. Instead, they relied on traditional information retrieval metrics like accuracy, precision, recall, and the F_1 score.

4.1.2 Unsupervised Metrics. We now discuss unsupervised approaches. In general, there are far fewer works relying on unsupervised metrics to evaluate the final narrative output.

Coherence. In general, coherence is not used to evaluate the output narrative despite being a useful metric during the extraction process. One exception is Xu and Tang [122], who evaluate their output using a weighted average of story coherence (based on Jaccard similarity) and story size.

Coverage. Xu and Tang [122] evaluate their output by treating coverage as a structural measure, making the assumption that good coverage of topics means that the structure of their metro map representation is good. However, due to the formulation of coverage based on whether the topical clusters of the dataset are covered, it does not explicitly consider the structure of the output, which makes it inappropriate to evaluate the structure. In contrast, Bögel and Gertz [14] use a notion of coverage based on event connections in a graph (i.e., an article is covered if there is at least one edge connecting it) that could be treated more as a structural measure than the topical concept of coverage.

Dispersion. Camacho Barranco et al. [18] use the *dispersion coefficient*—originally proposed as an evaluation metric for storytelling in the intelligence analysis domain [45]—to evaluate their storyline. In particular, the dispersion coefficient is based on the Soergel distance, although other distance metrics could be used [92]. In particular, dispersion is based on Swanson's complementary but disjoint hypothesis [104]—where articles that have no explicit common elements yield important inferences or insights when combined. These insights are not apparent from the separate documents. Furthermore, the authors propose a new evaluation metric to measure story flow based on Swanson's hypothesis called the *dispersion coefficient*, shown in Equation (3). We note that this particular version is based on the Soergel distance (S), but any other distance metric between documents could be used in practice.

$$\text{Dispersion}(d_1, \ldots, d_n) = 1 - \frac{1}{n-2} \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} \text{D}(d_i, d_j), \quad \text{with} \quad \text{D}(d_i, d_j) = \begin{cases} \frac{1}{n+i-j}, & \text{if S}(d_i, d_j) < \theta \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Diversity and Redundancy. Finally, another alternative is a diversity metric to ensure proper coverage or low redundancy. In particular, Duan et al. [28] used diversity—based on the average pairwise similarity of sentences (see Equation (4))—to evaluate the performance of their comparative timeline extraction method.

$$\text{Diversity} = 1 - \frac{1}{|S|^2} \sum s_i \in S \sum s_j \in S 1 - \text{CosineSim}(s_i, s_j) \tag{4}$$

4.2 User Evaluation Metrics

These metrics seek to evaluate the extracted narrative based on subjective user measures or task performance measures.

Task-Oriented Evaluation. Task-oriented metrics require designing a series of benchmark tasks to measure the number of correct answers, accuracy, how much time the users take to complete the task, or some other measure of correctness or quality. A few works use task-oriented evaluation metrics: metro maps [98, 99], information cartography [100, 101], and the SToRe system [64, 65]. These works rely on event-based representations, and all of them evaluate extraction methods as retrieval tools following a similar approach. In particular, there are *micro-knowledge* tasks that measure how the extracted narratives help users *retrieve information* faster and *macro-knowledge* tasks that measure how the extracted narratives help users *understand the big picture*.

For micro-knowledge tasks, all works create a series of simple retrieval questions such that the answers can be easily classified as right or wrong—for example, retrieving dates, facts, relevant entities, or the main event descriptors. Users are evaluated by measuring how many correct answers (i.e., accuracy) they can get in a fixed amount of time and the rate at which they answer these questions [64, 65, 98]. Another metric used at the micro-knowledge level is the ease of navigation, estimated by the number of documents that users clicked per correct answer [64, 65, 98].

For macro-knowledge, some form of summarization is used to evaluate the narratives. Shahaf et al. [98] asked users to create summaries based on different narrative representations and then used crowdsourcing to evaluate user preference over those summaries. However, these benchmark tasks do not go beyond basic retrieval and summarization. Tasks that require higher levels of knowledge and cognitive work (e.g., analysis tasks) are not covered by these evaluations. In general, the inherent difficulty of designing benchmark tasks that can be easily evaluated might be one of the reasons user-based evaluations of extraction methods usually rely on subjective ratings rather than task-oriented metrics.

Subjective Evaluation. Most of the works that rely on user evaluations use subjective measures (i.e., user perception metrics). These subjective metrics include concepts from usability, including criteria such as user preference [47, 51, 101], visual presentation [51], and ease of use [51, 64]. Other metrics include effectiveness as perceived by the users (e.g., perceived helpfulness or usefulness), satisfaction, and comprehensibility [64, 65, 109]. Alternatively perceived familiarity before and after using the extracted narrative can be a useful measure of usefulness [96].

Last, user-perceived quality is another widely used approach to evaluate extracted narratives. The user-perceived quality criteria mostly correspond to the quality criteria metrics defined before [18, 51, 96, 97, 109, 134, 135], including coherence, coverage, redundancy, relevance, dispersion, and similar variations (e.g., broadness). We note that these user perception metrics suffer a similar problem as their computational counterparts—they are fuzzy concepts that could be defined differently. This is further compounded by the subjective nature of these evaluations.

Other works rely on asking users whether they consider specific elements of the narrative as correct—for example, asking whether a specific connection is correct, whether the selected documents are relevant, whether a specific storyline is logically coherent, or about the number of coherent and relevant documents [18, 66, 67]. This is similar to traditional information retrieval metrics that rely on ground truth information. However, in this case, rather than using a previously defined gold standard, the accuracy measures are defined purely on subjective perceptions. Finally, another approach is to ask users to compare the ground truth with the output narrative—from potentially multiple methods—and rank them according to their preference based on their knowledge of the topic [61].

5 DISCUSSION

We now discuss our findings. We start by addressing the structural choices in narrative representation. Next, we address some of the challenges of extraction methods. Then, we turn our attention toward evaluation methods, including benchmark datasets, computational metrics, and user-based

evaluations. Afterward, we discuss practical applications of news narrative extraction. Finally, we discuss recent trends, open challenges, and potential research directions.

5.1 Narrative Structure

The choice of the core structure is an important aspect of narrative representation. Using a linear structure provides a simple approach to represent a narrative with a single storyline, but it does not appropriately model the nuances of narratives with multiple stories. In contrast, graph-based structures allow the modeling of different interactions between storylines (e.g., convergent and divergent stories) [51]. Linear representations are implicitly directed, but graph-based representations may or may not be directed. Directed graphs usually exploit the underlying temporal relationships to determine the direction of the connections between elements. When the connection between basic units is guided by temporal constraints, it naturally gives rise to directed acyclic graphs. Directed acyclic graphs provide the most flexibility while also accounting for the temporal nature of a narrative. However, not all directed graph models are acyclic, as some use specific types of relationships that allow the creation of cycles (e.g., same-event relations).

A representation that falls between linear and fully graph-based representations is the tree-based representation [66, 67, 130, 132, 133]. Such models allow for more flexible structures than linear representations. In particular, they are able to model story divergence (i.e., multiple storylines splitting off from the root or other nodes). Unlike graph-based models, they are not able to model story convergence (e.g., two stories joining into a final event), as that would break the tree structure. Tree-based structures have not been deeply explored in the literature and could provide an intermediate approach between linear and graph-based representations in terms of complexity, allowing easier understanding by users while retaining some flexibility. However, the inability to model story convergence might limit their applications.

5.2 Extraction Methods

Scalability and Computational Cost. Most extraction methods discussed in this survey suffer from issues when dealing with big data, as the processing pipelines are quite expensive in terms of computational power and they might not be easily parallelizable. One of the simplest methods to reduce computational cost is to filter the data beforehand. This turns the computational cost problem into an information retrieval problem, where the most relevant documents must be retrieved before extracting the narrative itself. Many methods assume that the data has been pre-filtered to a relevant set of news articles. Including a filtering step adds an additional element to the pipeline, thus increasing the risk of errors. Moreover, defining an adequate concept of relevance for this method might prove problematic in itself. Nevertheless, this provides a simple approach to mitigate the ever-increasing available amount of data.

Another approach is to deal with extraction in an online manner [66]. Most news narrative extraction methods are offline methods that analyze an entire set of news articles. However, extracting the stories in an online manner without disrupting the pre-existing structure would offer a computationally cheaper alternative. This is similar to the approach used by traditional TDT systems that sought to track the events of a topic in an online manner [4]. However, it would also require handling the structure of events associated with the narrative, which is not considered by traditional TDT.

Unified Metrics. One of the limitations of current approaches is that there are multiple versions of coherence and similar metrics. Coherence itself is an ill-defined term in practice and formalizing it in a computational or mathematical definition is a difficult task. The different definitions of coherence-like metrics focus on measuring different aspects of the narrative. Moreover, additional constraints can be considered to enforce coherence beyond numerical metrics (e.g., events sharing

Table 4. Benchmark Data in the TLS Works

Dataset	Source	URL
Timeline 17	[110]	https://github.com/complementizer/news-tls
Crisis	[109]	https://github.com/complementizer/news-tls
COVID-TLS	[55]	https://github.com/MorenoLaQuatra/SDF-TLS
TLS-COVID19	[83]	https://github.com/LIAAD/tls-covid19
Entities	[37]	https://github.com/complementizer/news-tls
MTLS Data	[129]	https://yiyualt.github.io/mtlsdata/

common entities). In general, a hybrid extraction approach that mixes multiple metrics (e.g., through a linear combination) and also includes such constraints might provide better results.

5.3 Evaluation Methods

Benchmark Datasets. In general, most works collect their own data or use a subset of a pre-existing news data repository. For example, some use datasets from TDT literature [20, 21, 76, 116], DUC/TAC conferences [23], or other general news repositories [24]. Most works do not publish their datasets. However, there is a subset of TLS works that have provided evaluation datasets that have been adopted in several works as benchmark data. We present these datasets in Table 4. These datasets are appropriate for the "events as sentences" resolution level that TLS uses. However, they do not provide a direct way to evaluate methods that use other resolution levels. Furthermore, we note that there are no such benchmark datasets for the other resolution levels of the narrative extraction tasks considered in this survey. The lack of appropriate benchmark data for the document-level and cluster-level resolutions makes comparing methods harder and makes replicability harder.

Computational Metrics Limitations. We note that most of the narratives discussed in this survey only consider the content (e.g., traditional information retrieval metrics) without accounting for the order nor the structure of the narrative. Some metrics consider ordering information, although only at a linear structure level. For example, story-level measures of coherence consider the connections between consecutive documents [96] or the dispersion coefficient that models story flow [18]. The ranking evaluation metrics also include some underlying notion of order; however, this notion is limited to a linear structure at best. The structural version of coverage based on event connections used by Bögel and Gertz [14] is based on local connections only, but it does not account for the full structure of the graph. Thus, current metrics for the narrative extraction task are unable to deal with complex narrative structures.

Given this limitation, it would be ideal to consider metrics that account for both order and structure to provide a proper evaluation of a narrative. For linear narratives, it would be sufficient to consider content and order, as the structure itself is fixed. An approach to solve this would be to consider a metric based on weighted edit distance [117] as it considers both the order of the elements and their contents (by defining weights according to event similarities or an adaptation of the previously discussed metrics). For non-linear narratives, a similar approach could use graph edit distance [2] with custom costs, as this metric would consider structure, order, and content.

However, the previous proposal would be supervised, as we would need a narrative against which to compare the output. Devising an unsupervised approach is a more challenging issue, particularly for graph-based narrative representations. One alternative is to attempt to extend coherence and dispersion measures to such graphs. For example, given a directed graph with a single starting event and a single ending event, it could be possible to compute all routes from start to end and obtain a weighted average of the coherence or dispersion of these routes. However, for more complex graph structures, it might be too costly in computational terms to do such computation. Designing an unsupervised evaluation metric remains an open challenge.

Benchmark Tasks and User Evaluations. User-based evaluations usually focus on subjective measurements rather than objective task performance. This is due to the lack of properly defined evaluation frameworks and benchmark tasks. Current approaches rely on micro-knowledge tasks, which are tasks focused on information retrieval and evaluate the number of correct answers (i.e., user accuracy) over time, and macro-knowledge tasks, which are tasks focused on summarization that indirectly evaluate the quality of the extracted narrative by measuring the quality of a user-generated summary. These approaches are limited and do not capture all the nuances associated with narrative sensemaking. Moreover, they do not cover more complex tasks beyond retrieval and summarization.

One possible solution would be to design more holistic evaluations based on different types of benchmark tasks. In particular, the use of Bloom's taxonomy [13] could provide a useful framework to define such tasks as seen in other sensemaking applications [16] or cognitive tasks in general [25]. Another possible solution would be to borrow the concept of insight-based evaluations [78]. Rather than focusing on benchmark tasks with specifically defined tasks and correct answers, the evaluation would be open-ended and would focus on analyzing the insights generated by the users.

5.4 Practical Applications

Event-based news narrative extraction has several practical applications beyond journalistic analysis tasks. Most of these applications seek to help with the issue of *information overload* in different contexts [101]. We briefly discuss some potential applications explored or mentioned in some of the reviewed works.

Disaster Management. Disaster management [122, 130, 132, 133] could benefit from using extraction approaches to keep track of disasters or other similarly negative incidents. In particular, to minimize losses caused by a disaster, one of the critical tasks in disaster management is to efficiently analyze and understand situation updates. Doing this requires effective methods to navigate a multitude of documents such as news or reports related to the disaster. Domain experts need to obtain condensed information about the disaster and its evolution [59]. Thus, news narrative extraction could help experts understand the evolving situation and devise a proper strategy.

Open Source Intelligence. **Open Source Intelligence (OSINT)** is intelligence that is synthesized using publicly available data [32]. Although OSINT data sources leverage more than just traditional news articles [38], OSINT could still benefit from news narrative extraction techniques. In particular, news narrative extraction methods could help intelligence analysts explore the information landscape and find key events [51]. Furthermore, these techniques could help analysts in prediction tasks by providing support and evidence [18].

Misinformation and Fact-Checking. News narrative extraction methods could aid fact-checkers in their tasks by providing them with an overview of the current narrative and highlighting key relevant events [51]. However, current methods do not include explicit ways to model misleading or outright false information.

Financial Markets. News narrative extraction could aid financial analysts to understand the information landscape [17]. For example, market news is regarded as an important data source in the context of financial analysis [17]. In particular, being able to understand and exploit the hidden information in the raw news data could help analysts adapt their strategies and reduce their financial risk.

5.5 Recent Trends and Open Challenges

TLS Variations. Recent works have proposed some variations on the traditional TLS task. In particular, Duan et al. [28] proposed the *comparative* TLS task and Yu et al. [129] proposed the MTLS

task. These two works highlight the fact that simple linear representations of narratives are naturally limiting unless applied to the most simple of narratives. Thus, the creation of similar tasks to address some of the shortcomings of these representations is a natural progression. However, it raises the question of whether these extensions would benefit from borrowing elements from the methods that use more complex representations discussed in this survey. A natural extension would be to consider a graph-based representation that allows for multiple storylines and comparisons without further modifications. This approach would address both the comparative TLS and MTLS tasks.

In this context, we note that most of the reviewed articles with a sentence-level event resolution used a linear structure (see Table 1). The only exceptions were the disaster storyline extraction systems [130, 132, 133] with their local tree representation. However, these methods are designed with a specific news topic in mind—disaster news—and are able to leverage specific characteristics of the topic (e.g., the disaster moves over time). Thus, it would not be possible to directly adapt it to other types of news without addressing this issue.

Furthermore, we note that there are no inherent limitations to sentence-level representations that prevent them from being extended beyond linear narratives, which makes the lack of graph-based approaches an opportunity for future research. Finally, although we did not find such a suitable graph-based approach in the traditional news domain, there is one example from the social media domain—which has its own set of challenges in terms of narrative extraction—that can be found in the work of Ansah et al. [6]. This work proposes a tree-based narrative representation with sentence-level event representation using tweets. This approach extends the traditional TLS by allowing divergent storylines to emerge instead of just a single timeline. Such an approach could be adapted to traditional news narrative extraction.

Multi-Resolution Methods. Currently, all the narrative extraction approaches that we reviewed work on a singular resolution level (sentences, documents, or clusters). Existing attempts at multiple resolution levels only change the scope of the data [100, 101] (i.e., applying the method again on a new subset of the data); they do not seek to change the underlying event resolution. Another perspective corresponds to the multi-level presentations of disaster storylines by Zhou et al. [132, 133] and Yuan et al. [130], which use global and local levels to represent the narrative. However, the underlying event representation remains the same, and no efforts have been made to make a model that handles multiple levels of event resolution. Developing models that provide a multi-resolution approach remains an open challenge.

Interactivity. Most works on news narrative extraction provide surface-level interactions [100, 101, 106] such as re-arranging elements and changing the layout, showing details on demand (e.g., all details about a news article), zooming, or performing basic filtering, highlighting, and searching. However, there is still a need for better interaction models that give users more control and feedback when exploring and manipulating the narrative. Some models [96, 97] allow more in-depth refinement by letting the user specify elements that need to be changed and then evaluating all possible replacement and insertion actions. Building upon this feature-based feedback, Shahaf et al. [98] designed a method to learn a *personalized coverage function* that can be optimized to find a personalized narrative.

Another approach by Bögel and Gertz [14] allows parametric interaction to modify the extracted graph in real time, helping the user understand how the narrative changes based on the parameters. However, this approach requires the users to understand the underlying model parameters. In this context, *semantic interactions* could be useful to aid users modify the model without deep understanding of the underlying parameters. Semantic interactions [118] are used in sensemaking applications to directly reflect the analytical thought process of analysts about data (e.g., by using information about how analysts organize documents or highlight text), as opposed to parametric

interaction that manipulates model parameters (e.g., sliders and keyword weights). Thus, capturing a user model through semantic interaction could lead to a better narrative model.

Misinformation in News. Recent works have highlighted the need for future work to model source bias, information validity, transparency, and credibility as an effort to model and counter misinformation [51, 56]. Existing narrative representations could be enhanced by including additional attributes in their representations and extraction algorithms.

Works on misinformation detection focus on the propagation structure and content to determine whether a certain article or publication contains misinformation [41, 120]. Other methods rely on crowdsourcing [41] to detect misinformative content early. However, these methods do not model misinformation as part of an overarching narrative. Instead, they focus on local elements (e.g., a specific event). Thus, a holistic narrative approach could be useful in this context.

The issue of misinformation is also highly relevant for a series of recent works on disaster tracking by using news narrative extraction [122, 130, 133]. However, none of these methods address this issue and rely on the underlying assumption that the set does not contain false or misleading information. Thus, creating a narrative extraction model that accounts for misinformation would be of vital importance in the context of disaster tracking.

6 CONCLUSION

This literature review focused on narrative extraction and its related tasks of representation and analysis, synthesizing findings from 54 studies and identifying recurring types of representational structures, extraction criteria, and evaluation metrics. We further analyzed the articles and identified a series of recent trends, open challenges, and potential research directions. In terms of limitations, we highlight the lack of benchmark datasets, the need for better evaluation metrics that are capable of handling complex narratives properly, the high computational costs of most methods, and the lack of standardized benchmark tasks for user-based evaluations. In terms of open challenges, we note the need for better interaction models that allow users to explore the narrative with more control. Finally, we note that current models do not handle misleading or false content, a rising challenge as misinformation compounds with information overload to make understanding the information landscape even harder.

As with other literature reviews, this work has some limitations related to the inclusion and exclusion of relevant pieces of work. In particular, we used the Scopus and Web of Science databases as our initial sources. Previous studies have shown that Scopus and Web of Science are inclusive and extensive sources for literature reviews [33]. Regardless, multiple studies were not included in our initial results, and thus we had to include them through other means, such as extracting relevant citations from reviewed works. Moreover, the choice of keywords might have caused some studies that use different terminology to not show up in our searches.

REFERENCES

[1] H. Porter Abbott. 2008. *The Cambridge Introduction to Narrative.* Cambridge University Press, New York, NY.

[2] Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *Proceedings of the 2015 4th International Conference on Pattern Recognition Applications and Methods.* 1–9.

[3] Arwa I. Alhussain and Aqil M. Azmi. 2021. Automatic story generation: A survey of approaches. *ACM Computing Surveys* 54, 5 (May 2021), Article 103, 38 pages.

[4] James Allan. 2012. *Topic Detection and Tracking: Event-Based Information Organization.* Vol. 12. Springer Science & Business Media, New York, NY.

[5] Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting Relevant News Events for Timeline Summaries. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13).* Association for Computing Machinery, 91–92.

[6] Jeffery Ansah, Lin Liu, Wei Kang, Selasie Kwashie, Jixue Li, and Jiuyong Li. 2019. A graph is worth a thousand words: Telling event stories using timeline summarization graphs. In *Proceedings of the World Wide Web Conference (WWW'19)*. ACM, New York, NY, 2565–2571.

[7] Eleftherios Avramidis. 2013. RankEval: Open tool for evaluation of machine-learned ranking. *Prague Bulletin of Mathematical Linguistics* 100 (2013), 63–72.

[8] Chris Baber, Dan Andrews, Tom Duffy, and Richard McMaster. 2011. Sensemaking as narrative: Visualization for collaboration. In *Proceedings of the 3rd International UKVAC Workshop on Visual Analytics (VAW'11)*. 7–8.

[9] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. 2010. Fast incremental and personalized PageRank. *Proceedings of the VLDB Endowment* 4, 3 (2010), 173–184.

[10] Kiran Kumar Bandeli, Muhammad Nihal Hussain, and Nitin Agarwal. 2020. A framework towards computational narrative analysis on blogs. In *Proceedings of the 3rd Workshop on Narrative Extraction from Texts Co-Located with the 42nd European Conference on Information Retrieval (Text2Story@ECIR'20)*. 63–69.

[11] David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, New York, NY, 113–120.

[12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (Jan. 2003), 993–1022.

[13] Benjamin Samuel Bloom. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longman, Ann Arbor, MI.

[14] Thomas Bögel and Michael Gertz. 2015. Time will tell: Temporal linking of news stories. In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, New York, NY, 195–204.

[15] Kenneth Burke. 1969. *A Grammar of Motives*. Vol. 177. University of California Press, Oakland, CA.

[16] Alyxander Burns, Cindy Xiong, Steven Franconeri, Alberto Cairo, and Narges Mahyar. 2020. How to evaluate data visualizations across different levels of understanding. In *Proceedings of the 2020 IEEE Workshop on Evaluation and Beyond: Methodological Approaches to Visualization*. IEEE, Los Alamitos, CA, 19–28.

[17] Yi Cai, Haoran Xie, Raymond Y. K. Lau, Qing Li, Tak-Lam Wong, and Fu Lee Wang. 2019. Temporal event searches based on event maps and relationships. *Applied Soft Computing* 85 (2019), 105750.

[18] Roberto Camacho Barranco, Arnold P. Boedihardjo, and M. Shahriar Hossain. 2019. Analyzing evolving stories in news articles. *International Journal of Data Science and Analytics* 8, 3 (2019), 241–256.

[19] Marc Cavazza and David Pizzi. 2006. Narratology for interactive storytelling: A critical introduction. In *Technologies for Interactive Digital Storytelling and Entertainment*, Stefan Göbel, Rainer Malkewitz, and Ido Iurgel (Eds.). Springer, Berlin, Germany, 72–83.

[20] Chien Chin Chen and Meng Chang Chen. 2008. TSCAN: A novel method for topic summarization and content anatomy. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. ACM, New York, NY, 579–586.

[21] Chien Chin Chen and Meng Chang Chen. 2012. TSCAN: A content anatomy approach to temporal topic summarization. *IEEE Transactions on Knowledge and Data Engineering* 24, 1 (2012), 170–183. https://www.computer.org/csdl/journal/tk/2012/01/ttk2012010170/13rRUwbJD5d.

[22] Chien Chin Chen, Yao-Tsung Chen, Yeali Sun, and Meng Chang Chen. 2003. Life cycle modeling of news events using aging theory. In *Machine Learning: ECML 2003*, Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski (Eds.). Springer, Berlin, Germany, 47–59.

[23] Jie Chen, Zhendong Niu, and Hongping Fu. 2015. A multi-news timeline summarization algorithm based on aging theory. In *Web Technologies and Applications*, Reynold Cheng, Bin Cui, Zhenjie Zhang, Ruichu Cai, and Jia Xu (Eds.). Springer, Cham, Switzerland, 449–460.

[24] Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. ACM, New York, NY, 425–432.

[25] Kevyn Collins-Thompson, Soo Young Rieh, Carl C. Haynes, and Rohail Syed. 2016. Assessing learning outcomes in web search: A comparison of tasks and query strategies. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*. ACM, New York, NY, 163–172.

[26] Samuel Y. Dennis III. 1991. On the hyper-Dirichlet type 1 and hyper-Liouville distributions. *Communications in Statistics: Theory and Methods* 20, 12 (1991), 4069–4081.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* (2018).

[28] Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. 2020. Comparative timeline summarization via dynamic affinity-preserving random walk. In *ECAI 2020*. IOS Press, Santiago de Compostela, Spain, 1778–1785.

[29] Gunes Erkan and Dragomir Radev. 2004. LexPageRank: Prestige in multi-document text summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 365–371.

[30] Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.
[31] Liana Ermakova, Jean Valère Cossu, and Josiane Mothe. 2019. A survey on evaluation of summarization methods. *Information Processing & Management* 56, 5 (2019), 1794–1814.
[32] João Rafael Gonçalves Evangelista, Renato José Sassi, Márcio Romero, and Domingos Napolitano. 2021. Systematic literature review to investigate the application of open source intelligence (OSINT) with artificial intelligence. *Journal of Applied Security Research* 16, 3 (2021), 345–369.
[33] Matthew E. Falagas, Eleni I. Pitsouni, George A. Malietzis, and Georgios Pappas. 2008. Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and weaknesses. *FASEB Journal* 22, 2 (2008), 338–342.
[34] Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.
[35] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.
[36] Pablo Gervás, Eugenio Concepción, Carlos León, Gonzalo Méndez, and Pablo Delatorre. 2019. The long path to narrative generation. *IBM Journal of Research and Development* 63, 1 (2019), Article 8, 10 pages.
[37] Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the state-of-the-art in news timeline summarization. *CoRR* abs/2005.10107 (2020).
[38] Helen Gibson. 2016. Acquisition and preparation of data for OSINT investigations. In *Open Source Intelligence Investigation: From Strategy to Implementation*, Babak Akhgar, P. Saskia Bayerl, and Fraser Sampson (Eds.). Springer, Cham, Switzerland, 69–93.
[39] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Jamie Callan. 2000. Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM'00)*. ACM, New York, NY, 165–172.
[40] R. Guha, Ravi Kumar, D. Sivakumar, and Ravi Sundaram. 2005. Unweaving a web of documents. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD'05)*. ACM, New York, NY, 574–579.
[41] Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2019. The future of misinformation detection: New perspectives and trends. *CoRR* abs/1909.03654 (2019).
[42] Jeffry Halverson, Steven Corman, and H. Lloyd Goodall. 2011. *Master Narratives of Islamist Extremism*. Springer, New York, NY.
[43] Taher H. Haveliwala. 2003. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering* 15, 4 (2003), 784–796.
[44] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, NY, 50–57.
[45] Mahmud Shahriar Hossain, Christopher Andrews, Naren Ramakrishnan, and Chris North. 2011. Helping intelligence analysts make connections. In *Proceedings of the 17th AAAI Conference on Scalable Integration of Analytics and Visualization (AAAIWS'11)*. 22–31.
[46] Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K. Usadi, and Xiaoyan Zhu. 2011. Generating breakpoint-based timeline overview for news topic retrospection. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. IEEE, Los Alamitos, CA, 260–269.
[47] Po Hu, Min-Lie Huang, and Xiao-Yan Zhu. 2014. Exploring the interactions of storylines from informative news events. *Journal of Computer Science and Technology* 29, 3 (2014), 502–518.
[48] Dongping Huang, Shuyu Hu, Yi Cai, and Huaqing Min. 2014. Discovering event evolution graphs based on news articles relationships. In *Proceedings of the 2014 IEEE 11th International Conference on e-Business Engineering*. IEEE, Los Alamitos, CA, 246–251.
[49] Lifu Huang and Lian'en Huang. 2013. Optimized event storyline generation based on mixture-event-aspect model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 726–735.
[50] Brian Keith Norambuena, Michael Horning, and Tanushree Mitra. 2020. Evaluating the inverted pyramid structure through automatic 5W1H extraction and summarization. In *Proceedings of the 2020 Computation + Journalism Symposium*. 1–7.
[51] Brian Felipe Keith Norambuena and Tanushree Mitra. 2021. Narrative maps: An algorithmic approach to represent and extract information narratives. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–33.
[52] Arpit Khurdiya, Lipika Dey, Nidhi Raj, and Sk. Mirajul Haque. 2011. Multi-perspective linking of news articles within a repository. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. 2281–2286.
[53] Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics* 22, 1 (1951), 79–86.

[54] Ben Kybartas and Rafael Bidarra. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 3 (2016), 239–253.
[55] Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. Summarize dates first: A paradigm shift in timeline summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'21)*. ACM, New York, NY, 418–427.
[56] Philippe Laban and Marti A. Hearst. 2017. newsLens: Building and visualizing long-ranging news stories. In *Proceedings of the Events and Stories in the News Workshop*. 1–9.
[57] Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys* 52, 5 (Sept. 2019), Article 89, 40 pages.
[58] Jiwei Li and Sujian Li. 2013. Evolutionary hierarchical Dirichlet process for timeline summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 556–560.
[59] Lei Li and Tao Li. 2013. An empirical study of ontology-based multi-document summarization in disaster management. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, 2 (2013), 162–171.
[60] Rumeng Li, Tao Wang, and Xun Wang. 2015. *Tracking Events Using Time-Dependent Hierarchical Dirichlet Tree Model*. 550–558.
[61] Yiming Liao, Shuguang Wang, and Dongwon Lee. 2021. WILSON: A divide and conquer approach for fast and effective news timeline summarization. In *Advances in Database Technology—EDBT*, Yannis Velegrakis, Yannis Velegrakis, Demetris Zeinalipour, Panos K. Chrysanthis, Panos K. Chrysanthis, and Francesco Guerra (Eds.). OpenProceedings.org, Nicosia, Cyprus, 635–645.
[62] Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. ACM, New York, NY, 175–184.
[63] Fu-Ren Lin, Feng-Mei Huang, and Chia-Hao Liang. 2007. Individualized storyline-based news topic retrospection. In *PACIS 2007 Proceedings* AIS, Auckland, New Zealand, 140.
[64] Fu-Ren Lin and Chia-Hao Liang. 2006. Topic retrospection with storyline-based summarization on news reports. In *PACIS 2006 Proceedings*. AIS, Kuala Lumpur, Malaysia, 1320–1334.
[65] Fu-Ren Lin and Chia-Hao Liang. 2008. Storyline-based summarization for news topic retrospection. *Decision Support Systems* 45, 3 (2008), 473–490.
[66] Bang Liu, Fred X. Han, Di Niu, Linglong Kong, Kunfeng Lai, and Yu Xu. 2020. Story Forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data* 14, 3 (2020), 1–28.
[67] Bang Liu, Di Niu, Kunfeng Lai, Linglong Kong, and Yu Xu. 2017. Growing Story Forest online from massive breaking news. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. ACM, New York, NY, 777–785.
[68] Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2, 2 (1958), 159–165.
[69] Inderjeet Mani. 2012. Computational modeling of narrative. *Synthesis Lectures on Human Language Technologies* 5, 3 (2012), 1–142.
[70] Qiaozhu Mei, Jian Guo, and Dragomir Radev. 2010. DivRank: The interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 1009–1018.
[71] Qiaozhu Mei and ChengXiang Zhai. 2005. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD'05)*. ACM, New York, NY, 198–207.
[72] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*. 775–780.
[73] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 404–411.
[74] Anne-Lyse Myriam Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. Semeval-2015 Task 4: Timeline: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. 778–786.
[75] Alister Miskimmon, Ben O'Loughlin, and Laura Roselle. 2014. *Strategic Narratives: Communication Power and the New World Order*. Routledge, New York, NY.
[76] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. 2004. Event threading within news topics. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM'04)*. ACM, New York, NY, 446–453.
[77] Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 1208–1217.

[78] Chris North, Purvi Saraiya, and Karen Duca. 2011. A comparison of benchmark task and insight evaluation methods for information visualization. *Information Visualization* 10, 3 (2011), 162–181.
[79] Nir Ofek, Sándor Darányi, and Lior Rokach. 2013. Linking motif sequences with tale types by machine learning. In *Proceedings of the 2013 Workshop on Computational Models of Narrative*. 166–182.
[80] Takashi Ogata. 2016. Computational and cognitive approaches to narratology from the perspective of narrative generation. In *Computational and Cognitive Approaches to Narratology*. IGI Global, Hershey, PA, 1–74.
[81] Yasuhiro Ogawa, Masaki Mori, and Katsuhiko Toyama. 2012. Recall-oriented evaluation metrics for consistent translation of Japanese legal sentences. In *New Frontiers in Artificial Intelligence*, Manabu Okumura, Daisuke Bekki, and Ken Satoh (Eds.). Springer, Berlin, Germany, 141–154.
[82] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab, Stanford, CA.
[83] Arian Pasquali, Ricardo Campos, Alexandre Ribeiro, Brenda Santana, Alípio Jorge, and Adam Jatowt. 2021. TLS-Covid19: A new annotated corpus for timeline summarization. In *Advances in Information Retrieval*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer International Publishing, Cham, Switzerland, 497–512.
[84] Robin Pemantle. 1992. Vertex-reinforced random walk. *Probability Theory and Related Fields* 92, 1 (1992), 117–136.
[85] Kent Puckett. 2016. *Narrative Theory*. Cambridge University Press, New York, NY.
[86] 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. *New Directions in Question Answering* 3 (2003), 28–34.
[87] Jiangtao Qiu, Chuan Li, Shaojie Qiao, Taiyong Li, and Jun Zhu. 2008. Timeline analysis of web news events. In *Advanced Data Mining and Applications*, Changjie Tang, Charles X. Ling, Xiaofang Zhou, Nick J. Cercone, and Xue Li (Eds.). Springer, Berlin, Germany, 123–134.
[88] J. Qiu and C. Tang. 2007. Topic oriented semi-supervised document clustering. In *Proceedings of the SIGMOD 2007 Workshop on Innovative Database Research (IDAR'07)*. 57–62.
[89] Jiangtao Qiu, Changjie Tang, Tao Zeng, Shaojie Qiao, Jie Zuo, Peng Chen, and Jun Zhu. 2007. A novel text classification approach based on enhanced association rule. In *Advanced Data Mining and Applications*, Reda Alhajj, Hong Gao, Jianzhong Li, Xue Li, and Osmar R. Zaïane (Eds.). Springer, Berlin, Germany, 252–263.
[90] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *CoRR* abs/1908.10084 (2019).
[91] Whitman Richards, Patrick Henry Winston, and Mark Alan Finlayson. 2009. *Advancing Computational Models of Narrative*. Technical Report. MIT-CSAIL.
[92] J. T. Rigsby and Daniel Barbará. 2018. Storytelling with signal injection: Focusing stories with domain knowledge. In *Machine Learning and Data Mining in Pattern Recognition*, Petra Perner (Ed.). Springer, Cham, Switzerland, 425–439.
[93] Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 410–420.
[94] James Ryan. 2017. Grimes' Fairy Tales: A 1960s story generator. In *Interactive Storytelling*, Nuno Nunes, Ian Oakley, and Valentina Nisi (Eds.). Springer, Cham, Switzerland, 89–103.
[95] Jason Schlachter, Alicia Ruvinsky, Luis Asencios Reynoso, Sathappan Muthiah, and Naren Ramakrishnan. 2015. Leveraging topic models to develop metrics for evaluating the quality of narrative threads extracted from news stories. *Procedia Manufacturing* 3 (2015), 4028–4035.
[96] Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 623–632.
[97] Dafna Shahaf and Carlos Guestrin. 2012. Connecting two (or less) dots: Discovering structure in news articles. *ACM Transactions on Knowledge Discovery from Data* 5, 4 (2012), 1–31.
[98] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: Generating information maps. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, New York, NY, 899–908.
[99] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2013. Metro maps of information. *SIGWEB Newsletter* 2013, Spring (April 2013), Article 4, 9 pages.
[100] Dafna Shahaf, Carlos Guestrin, Eric Horvitz, and Jure Leskovec. 2015. Information cartography. *Communications of the ACM* 58, 11 (2015), 62–73.
[101] Dafna Shahaf, Jaewon Yang, Caroline Suen, Jeff Jacobs, Heidi Wang, and Jure Leskovec. 2013. Information cartography: Creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1097–1105.
[102] Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. 984–992.

[103] Russell Swan and James Allan. 2000. Automatic generation of overview timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*. ACM, New York, NY, 49–56.

[104] Don R. Swanson. 1991. Complementary structures in disjoint science literatures. In *Proceedings of the 4th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'91)*. ACM, New York, NY, 280–289.

[105] Joanna Szostek. 2017. Defence and promotion of desired state identity in Russia's strategic narrative. *Geopolitics* 22, 3 (2017), 571–593.

[106] Xavier Tannier and Véronique Moriceau. 2013. Building event threads out of multiple news articles. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 958–967.

[107] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101, 476 (2006), 1566–1581.

[108] Mikhail Tikhomirov and Boris Dobrov. 2018. News timeline generation: Accounting for structural aspects and temporal nature of news stream. In *Data Analytics and Management in Data Intensive Domains*, Leonid Kalinichenko, Yannis Manolopoulos, Oleg Malkov, Nikolay Skvortsov, Sergey Stupnikov, and Vladimir Sukhomlin (Eds.). Springer, Cham, Switzerland, 267–280.

[109] Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *Advances in Information Retrieval*, Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr (Eds.). Springer, Cham, Switzerland, 245–256.

[110] Giang Binh Tran, Tuan Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. Leveraging learning to rank in an optimization framework for timeline summarization. In *Proceedings of the SIGIR 2013 Workshop on Time-Aware Information Access (TAIA'13)*. ACM, New York, NY, 4.

[111] Naohiko Uramoto and Koichi Takeda. 1998. A method for relating multiple newspaper articles by using graphs, and its application to webcasting. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics—Volume 2 (ACL'98/COLING'98)*. 1307–1313.

[112] Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2017. From computational narrative analysis to generation: A preliminary review. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG'17)*. ACM, New York, NY, Article 55, 4 pages.

[113] Ellen M. Voorhees. 1986. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management* 22, 6 (1986), 465–476.

[114] Paul Wake. 2013. Narrative and narratology. In *The Routledge Companion to Critical and Cultural Theory*. Routledge, New York, NY, 39–52.

[115] Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2018. Event phase oriented news summarization. *World Wide Web* 21, 4 (2018), 1069–1092.

[116] Chih-Ping Wei, Yen-Hsien Lee, Yu-Sheng Chiang, Chun-Ta Chen, and Christopher C. C. Yang. 2014. Exploiting temporal characteristics of features for effectively discovering event episodes from news corpora. *Journal of the Association for Information Science and Technology* 65, 3 (2014), 621–634.

[117] Achim Weigel and Frank Fein. 1994. Normalizing the weighted edit distance. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Volume 3—Conference C: Signal Processing*. IEEE, Los Alamitos, CA, 399–402.

[118] John Wenskovitch, Lauren Bradel, Michelle Dowling, Leanna House, and Chris North. 2018. The effect of semantic interaction on foraging in text analysis. In *Proceedings of the 2018 IEEE Conference on Visual Analytics Science and Technology (VAST'18)*. IEEE, Los Alamitos, CA, 13–24.

[119] Tom Wilson, Kaitlyn Zhou, and Kate Starbird. 2018. Assembling strategic narratives: Information operations as collaborative work within an online community. *Proceedings of the ACM on Human Computer Interaction* 2, CSCW (2018), Article 183, 26 pages.

[120] Liang Wu, Fred Morstatter, Kathleen M. Carley, and Huan Liu. 2019. Misinformation in social media: Definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter* 21, 2 (2019), 80–90.

[121] Yaguang Wu, Haichun Sun, and Chungang Yan. 2017. An event timeline extraction method based on news corpus. In *Proceedings of the 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA'17)*. IEEE, Los Alamitos, CA, 697–702.

[122] Nuo Xu and Xijin Tang. 2018. Generating risk maps for evolution analysis of societal risk events. In *Knowledge and Systems Sciences*, Jian Chen, Yuji Yamada, Mina Ryoke, and Xijin Tang (Eds.). Springer, Singapore, 115–128.

[123] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 433–443.

[124] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 745–754.

[125] Christopher C. Yang, Xiaodong Shi, and Chih-Ping Wei. 2006. Tracing the event evolution of terror attacks from on-line news. In *Intelligence and Security Informatics*, Sharad Mehrotra, Daniel D. Zeng, Hsinchun Chen, Bhavani Thuraisingham, and Fei-Yue Wang (Eds.). Springer, Berlin, Germany, 343–354.

[126] Christopher C. Yang, Xiaodong Shi, and Chih-Ping Wei. 2009. Discovering event evolution graphs from news corpora. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* 39, 4 (2009), 850–863.

[127] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. ACM, New York, NY, 603–610.

[128] Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. 129–136.

[129] Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. Multi-timeline summarization (MTLS): Improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 377–387.

[130] Ruifeng Yuan, Qifeng Zhou, and Wubai Zhou. 2019. dTexSL: A dynamic disaster textual storyline generating framework. *World Wide Web* 22, 5 (2019), 1913–1933.

[131] Liang Zhao. 2021. Event prediction in the big data era: A systematic survey. *ACM Computing Surveys* 54, 5 (2021), 1–37.

[132] Wubai Zhou, Chao Shen, Tao Li, Shu-Ching Chen, and Ning Xie. 2014. Generating textual storyline to improve situation awareness in disaster management. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration*. IEEE, Los Alamitos, CA, 585–592.

[133] Wubai Zhou, Chao Shen, Tao Li, Shu-Ching Chen, Ning Xie, and S. S. Iyengar. 2018. A new two-layer storyline generation framework for disaster management. *International Journal of Next-Generation Computing* 9, 3 (2018), 13.

[134] Xianshu Zhu and Tim Oates. 2012. Finding story chains in newswire articles. In *Proceedings of the 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI'12)*. IEEE, Los Alamitos, CA, 93–100.

[135] Xianshu Zhu and Tim Oates. 2014. Finding story chains in newswire articles using random walks. *Information Systems Frontiers* 16, 5 (2014), 753–769.

Received 16 September 2021; revised 2 May 2022; accepted 14 February 2023