LLM-Based Dependency Tracking for Short Event Descriptions

Naoki Sawahata[1], Daisuke Machizawa[1], Ryohei Ikejiri[2], and Yasunobu Sumikawa[1]

[1] Takushoku University, Tokyo, Japan sumikawa.lab@gmail.com
[2] Hiroshima University, Hiroshima, Japan
rikejiri@hiroshima-u.ac.jp

Abstract. Structuring the relationships through which events occur is crucial for gaining a deeper understanding of the past and for applying that knowledge to the present. In this study, we propose a method for constructing event networks that represent dependencies between events using large language models (LLMs). Our method performs both network and node selection to identify event pairs prior to applying the LLM. To evaluate the method, we collected past event data from Wikipedia and created a network dataset. An evaluation of our method showed an F1 score of 45.1%, demonstrating its effectiveness.

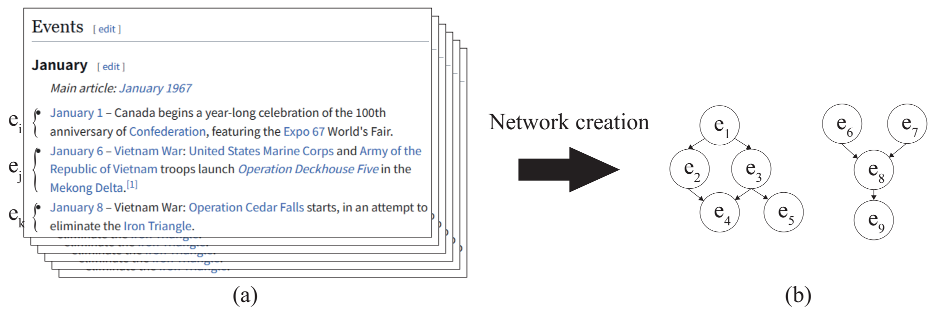
Keywords: Wikipedia· events · event graph · dependency · causal relationship

1 Introduction

Many digital libraries, such as Wikipedia, record events that have occurred throughout history. These events are often arranged in chronological order based on their dates of occurrence, with the dependencies between them typically left unspecified, as shown in Fig. 1 (a). As the saying goes, *History doesn't repeat itself, but it often rhymes,* properly understanding the past provides important benefits, such as deepening our understanding of the formation processes of modern society and laying the foundation for developing strategies to address contemporary issues. Since digital libraries archive not only modern events but also those dating back to ancient times, structuring the dependencies among these events could serve as a crucial basis for understanding historical developments and providing enriched information access.

Various methods have been proposed to construct timelines [6,8,17,31], tree structures [14,32], and networks [30] from lengthy event descriptions found in news articles. However, as shown in Fig. 1 (a), the events recorded in digital libraries are often described so briefly that extracting contextual information from the text itself becomes difficult, making it challenging for existing methods to construct appropriate networks.

In this study, we propose a method for structuring dependencies between events described in short texts, where contextual information is often limited.


Events [edit]

January [edit]

Main article: January 1967

- January 1 – Canada begins a year-long celebration of the 100th anniversary of Confederation, featuring the Expo 67 World's Fair.
- January 6 – Vietnam War: United States Marine Corps and Army of the Republic of Vietnam troops launch *Operation Deckhouse Five* in the Mekong Delta.[1]
- January 8 – Vietnam War: Operation Cedar Falls starts, in an attempt to eliminate the Iron Triangle.

Fig. 1. (a) Examples of events recorded in Wikipedia. (b) Example results of StoryNetworks.

Our approach represents these events as network graphs, which we refer to as StoryNetworks. Figure 1 (b) presents example StoryNetworks constructed using the proposed method. To analyze event dependencies despite the absence of explicit contextual cues, we leverage large language models (LLMs) to infer implicit relationships, such as geopolitical associations or temporal overlaps, often suggested by named entities, e.g., countries or individuals, mentioned in the event descriptions. Accordingly, this study addresses the following research questions:

- Are LLMs effective at analyzing dependencies between events described in short texts?
- How can LLMs be utilized to construct a network of event dependencies?

To mitigate the introduction of spurious or redundant edges in the StoryNetwork due to overly broad event pairings, our method performs two-step selection algorithms: (1) grouping semantically similar events, and (2) selecting specific pairs of events within each group for targeted dependency analysis using LLMs.

To the best of our knowledge, no existing dataset consists solely of networks constructed from short event descriptions. Therefore, in this study, we manually constructed a dataset[3] by collecting short-text event data from the Current events portal of Wikipedia[4] that provides daily updates on notable global news and recent events. Using this dataset, we conducted experiments comparing our proposed method with previous methods, which assume the availability of longer texts. The results showed that while the baseline methods achieved an F1 score of 27.9%, our method achieved a significantly higher score of 45.1%, demonstrating its superior performance.

The contributions of this study are summarized as follows:

- We tackled a novel problem of constructing event networks from short-text event descriptions.
- We created a new event network dataset that includes a larger number of events.

[3] Our dataset is available at `https://github.com/sumilab/dataset`.
[4] `https://en.wikipedia.org/wiki/Portal:Current_events`

– We designed an algorithm for constructing event networks and demonstrated its effectiveness through experimental evaluation.

2 Related Work

Previous research efforts aimed at structuring related events can be broadly categorized into three main areas: topic detection and tracking (TDT), timeline summarization (TLS), and event evolution graph (EEG) creation.

TDT focuses on identifying which previously reported news articles are related to newly reported ones. Research in this area includes studies aimed at predicting future developments [21] and detecting trending events on social media platforms [11,15,20,24]. Traditionally, TDT has been achieved by measuring textual similarity based on the bag-of-words model and applying clustering algorithms to detect events that belong to the same topic [10,13]. In addition, there have been proposals to realize TDT using probabilistic clustering models [3,7,29] as well as methods based on BERT [28]. These studies primarily propose methods for arranging related events in chronological order. More recently, research has extended beyond linear arrangement by representing events in a tree structure to clarify the underlying timelines they form [14,32].

From a graph-structural perspective, a key difference between our method and those proposed in previous research lies in the ability to generate *event joining*, where multiple events collectively depend on a single subsequent event. Specifically, while our proposed method is capable of generating such event joining, prior approaches are not designed to do so. As stated in Section 1, in order to gain a deeper understanding of the past, it is important not only to list events chronologically as a timeline but also to clearly express which events could actually serve as causes, as represented through event joining.

Timeline summarization focuses not simply on listing related events but on selecting key events, words [22], or sentences [5] to facilitate easier understanding, and arranging them in chronological order. According to [8], methods for timeline summarization can be broadly classified into three types: direct summarization, date-wise approach, and event detection. Direct summarization generates a timeline by selecting multiple sentences [5,17,19]. Date-wise approach selects sentences from the dataset based on the number of days and sentences per day, specified as hyperparameters, and compiles a timeline summary [18]. Event detection involves identifying events within the dataset and generating summaries using the most important events [8]. While most of these methods use long documents such as news articles to create timelines, other approaches have been proposed for microblogs, where the text is much shorter and noisier [2,23]. In methods targeting microblogs, features beyond simple textual similarity, such as the number of posts referencing the same event, are also incorporated into the timeline generation process. Additionally, there are approaches that use abstract meaning representation results to select sentences for timeline creation [17].

The timelines generated by the aforementioned methods are structured as a single chronological sequence of sentences per query. As a result, these methods fail to capture not only event joining but also event threading, a structure in which multiple events depend on a single preceding event. Although methods have been proposed that generate multiple timelines in response to a query, thereby representing different stories [6, 31], these approaches still produce multiple linear sequences and are not capable of directly representing dependency relationships where multiple events converge into a single event.

Research aiming to represent event dependencies as networks, similar to our study, includes the EEG proposed by [30]. This previous study proposed a method for automatically constructing networks using an extension of TF-IDF and cosine similarity. The key differences between the previous study and our work are as follows: 1) Length of event descriptions: our study assumes that each event is described in a short text, whereas EEG assumes the availability of longer, more detailed news articles. 2) Assumptions about evolution: previous studies assume that events related through evolution share similar vocabulary or key terms. However, when dealing with short event descriptions, it is often the case that events primarily share information related to countries or regions. This can bias results toward clustering events geographically. In contrast, our study leverages LLMs to extract important vocabulary and key concepts beyond geographic or national references. 3) Method of capturing contextual information: EEG extracts context only from within the news articles themselves, and rely on related news lists available on CNN News websites during dataset construction. In contrast, we utilize LLMs to infer contextual information, even when it is not explicitly stated in the text. 4) Dataset construction: [30] collected 176 events. In comparison, we collected over 5,000 event texts from Wikipedia, which is approximately 28 times larger than that of the previous study.

Studies on dependency extraction using LLMs have also been reported [16]. However, even in this study, the target texts are long documents such as news articles, as the objective is to extract event pairs within a single document.

3 Dataset Creation

In this section, we describe the data collection process, followed by the procedures for constructing the dataset from the collected data.

3.1 Data Collection

We utilized Wikipedia as the primary source of information for data collection as it contains a vast amount of recorded events suitable for constructing various types of networks. Wikipedia records events dating from ancient times to the present, organized by each year[5] and each month and day[6]. Manually constructing networks from all of this data would incur a substantial cost; thus, this study created seed networks using events recorded in the Wikipedia current portal from January 1, 2016, to December 31, 2017.

[5] e.g., `https://en.wikipedia.org/wiki/2020`
[6] e.g., `https://en.wikipedia.org/wiki/January_23`

Table 1. Dataset statistics.

Number of events used in timelines	5,204
Number of timelines	2,889
Number of dependencies	2,494
Period of timestamps	1 Jan. 2016 – 31 Dec. 2017

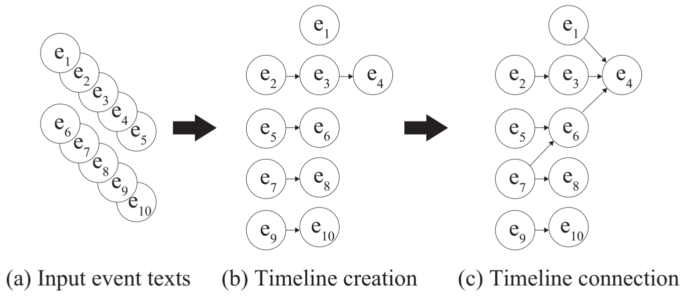


Fig. 2. Dataset creation process

Table 1 presents the statistics of the past event data collected in this study, along with the timelines and networks constructed from them.

3.2 Timeline Creation

Figure 2 illustrates the procedure used to construct the dataset in this study. The networks allow for multiple paths to emerge from an initial event, where each path can be regarded as a timeline. By connecting a single event to multiple subsequent events as its children, multiple distinct paths can be defined. In this structure, although sibling events share a certain degree of relatedness, they can be considered to belong to different topics. Therefore, we define a timeline as a sequence of events that are similar in content and share the same topic, as shown in Fig. 2 (b).

As discussed in Section 2, many studies have been conducted on timeline generation. This study adopts the position of utilizing timelines, rather than proposing a new method for generating them. Therefore, we first evaluated whether existing timeline generation methods can produce high-quality timelines from short event descriptions. To conduct this evaluation, we manually created 42 timelines using only the event data from January 2016. This task was performed by two annotators, and only timelines on which both annotators agreed were included. Among existing methods, we implemented the method proposed by Kira Radinsky [21], which utilizes textual information within event sentences, and evaluated its precision, recall, and F1 score, obtaining 16.6%, 9.5%, and 12.1%, respectively. These results indicate that the method is insufficient for constructing the dataset. Although methods using BERT or LLMs [16] have also been proposed, even when applied to longer texts, their F1 scores are only around

60%, which is still inadequate for reliable dataset construction. Consequently, in this study, we manually created the timelines.

The procedure for manual creation involved two annotators independently constructing timelines for the event data from 2016 and 2017, respectively. All of the timelines created in this process were then evaluated for validity by a third annotator. If any events needed to be added or removed, the third annotator discussed these changes with the original creators of the timelines and revised them accordingly. We created 2,889 timelines from the two years of events.

3.3 Timelines Connection

After manually verifying the timelines obtained through the procedure described in Section 3.2, it was found that even events which formed a dependency relationship, such as "The Ukraine war (war category) caused a rise in oil prices (economic category)," were assigned to different groups due to differing main topics. Particularly relevant events were manually extracted from multiple timelines and subsequently linked to construct a network, as shown in Fig. 2 (c). By identifying this dependency and drawing directed edges between events across different timelines, it becomes possible to graphically represent event threading and event joining.

For the created timelines, all possible combinations were generated, and the existence of dependency relationships between events of them was manually verified. The process of connecting the timelines was performed using the same procedure as timeline creation; two workers independently created connections for the timelines from 2016 and 2017. Subsequently, a third worker evaluated the validity of all the connected timelines created. If any additions or deletions of connections were necessary, discussions were held with the creators of the respective connected timelines to modify the connections accordingly. As a result of this process, 2,494 dependencies from networks were defined.

4 StoryNetwork Creation Algorithm

Our algorithm performs network selection and event selection to prevent the definition of excessive edges in the network construction. When humans construct a network, if multiple timelines are derived from a single event, edges between events within these derived timelines are defined only to the minimum necessary. In contrast, when an LLM is applied to identify relationships between two events, it may define edges excessively, even in cases where a human would not.

Algorithm 1 presents the main implementation steps of the proposed method. First, a list N is defined to manage all the networks to be created by this method (line 2). In line 3, the event e is processed one by one to either create a new network or expand an existing one. The function *NetworkSelect* in line 4 selects the network with the highest probability of including e from the existing networks. The function *FrontierEventSelect* in line 5 selects a set of candidate events Evt' for analyzing the connection with e from the events contained in n. The function

Algorithm 1 StoryNetwork creation

```
Input: Event set: $Evt$

1:  Function $SNetCreation$ $(Evt)$
2:    $N \leftarrow \emptyset$
3:    foreach $e \in Evt$
4:      $n \leftarrow NetworkSelect(N, e)$
5:      $Evt' \leftarrow FrontierEventSelect(e, n)$
6:      $EdgeList \leftarrow EdgeCreation(Evt', e, n)$
7:      $Update(N, n, EdgeList)$
8:  Function $EdgeCreation$ $(Evt', e, n)$
9:    $EdgeList \leftarrow []$
10:   foreach $e' \in Evt'$
11:     if $Prompt(e', e) = \text{Yes}$
12:       $EdgeList.\text{append}((e', e))$
13:   return $EdgeList$
```

EdgeCreation in line 6 analyzes the relationship between e and $e' \in Evt'$ and outputs a list of edges that are defined as a result. Finally, this list of edges is reflected in the network by the function *Update* in line 7. If N is an empty set, both n and Evt' will also be empty. In *EdgeCreation*, e will then be recorded as the unique node of n. Additionally, if *NetworkSelect* results in no existing network that should include e, n will be output as an empty set, and the subsequent process will proceed as described above. The detail procedures of *EdgeCreation* are defined in lines 9 to 13. First, the list of edges returned by this function is defined in line 9. In line 10, one event at a time is extracted from Evt', and whether to define an edge between it and e is determined by executing the function *Prompt*, which runs the LLM prompt.

In the following, we describe details of *NetworkSelect*, *FrontierEventSelect*, and *Prompt* in that order.

4.1 Network Selection

The network generated by this method varies depending on the content of the events; therefore, several networks may be constructed at the same time. The process begins by determining whether a newly input event e should be incorporated into an existing network n.

We executed a prompt on GPT-4o to determine which of the predefined groups a given event should belong to. The prompt instructed the model to return the corresponding group ID if a suitable group existed, or to define a new group if none were appropriate.

4.2 Frontier Event Selection

The primary objective of this step is to identify, for each event, a pairing with the most recent previously analyzed event that shares one or more timelines. We

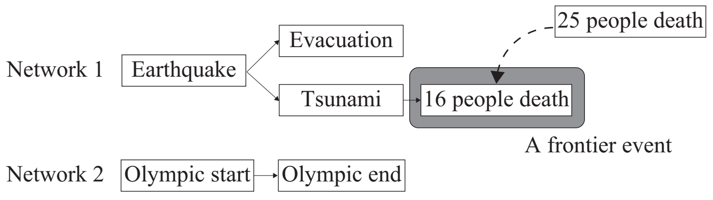


Fig. 3. Example of frontier event selection. The gray rectangles indicate frontier events.

refer to these most recent events as frontier events. The purpose of identifying this frontier event is to narrow down the scope of dependency analysis to be conducted in the subsequent step. This narrowing is carried out by determining whether the topics of the event descriptions are consistent. For instance, as shown in Fig. 3, multiple reports on the number of deaths caused by an earthquake share the same topic. When such updated information is reported, it suffices to identify that an update has occurred; therefore, only the most recent event among past similar events is selected. In this example, the new event `25 people death` should be only linked to `16 people death` as this event reports that the death toll has been revised. If we skip this selection step, the event could also be linked to `Earthquake` or `Tsunami`, resulting in an unnecessarily dense network structure. The objective of this step is formally defined as follows:

$$\mathit{FrontierEvents}(e) \overset{def}{\Leftrightarrow} \{e_j | (e_i, e_j) \in E(n), \mathit{isFront}(e, e_i, e_j)\} \tag{1}$$

where E is a function that returns all edges defined in the group n. *isFront* determines whether e_j is a frontier node of group n.

The formal definition of *isFront* is presented below. Let e and (e_i, e_j) be the newly analyzed event and the previously defined event pair, respectively. In this notation, it is assumed that the event written on the right occurred more recently than the one on the left; that is, e_j is assumed to be more recent than e_i. First, we analyze whether e and e_j share the same topic. To do this, we compute the topic distribution using BERTopic [9] and calculate the Euclidean distance between their feature vectors. If the Euclidean distance between the topic distribution scores is below a certain threshold, we conclude that the two events share the same topic. The formal definition of this calculation is given as follows:

$$\mathit{isSim}(e_k, e_l) \overset{def}{\Leftrightarrow} \mathit{Dist}(\mathit{FVec}(e_k), \mathit{FVec}(e_l)) < \mathit{Thre} \tag{2}$$

where *Dist* is a function that computes the Euclidean distance between two feature vectors. *FVec* applies BERTopic to generate a feature vector for the given event. *Thre* is a hyperparameter, which we set to 0.6. This value was selected based on the experimental results that yielded the best performance using the dataset created in this study. If these events share the same topic, we then analyze whether e and e_i also belong to the same topic. If this pair also shares the same topic, e_j is considered the frontier event, and we skip defining

* Background *
You are working for creating a dataset that includes dependencies or known as stories of events. This dataset helps us to know what events a single story has. Thus, if there is dependency, correlation, or chains as story between two events, this work sets a dependency between the two events.
* Task description *
I would like you to support creating the dataset. Could you judge whether there is a same story between the following two events? If your analysis finds any story, please say Yes. Otherwise, say No. Please say only Yes or No.
* Event 1:
<event1>
* Event 2:
<event2>

Fig. 4. Prompt template for ZS

the pair (e_i, e), defining only the pair (e_j, e). The following provides formal definitions of these processes.

$$isFront(e, e_i, e_j) \overset{def}{\Leftrightarrow} isSim(e, e_j) \land isSim(e, e_i) \tag{3}$$

If there are no shared topics between e and e_j, we then perform these analyses for other edges.

4.3 Edge Creation

We utilize LLM to determine whether an edge should be defined between two events. To implement this, three prompting methods are employed: zero-shot (ZS) prompting, few-shot (FS) prompting, and the combination of FS with Chain-of-Thought (CoT) prompting.

ZS prompting. ZS prompting refers to a method that does not provide examples during inference [4]. Our method presents an LLM with two event descriptions and ask whether a dependency exists between them. Importantly, no additional examples of dependencies or specific data characteristics are provided. If the LLM determines that a dependency exists, it is instructed to return "Yes"; otherwise, it returns "No".

Fig. 4 presents the prompt template created for this mode. The placeholders <event1> and <event2> are to be filled with the two event descriptions whose dependency is to be examined.

FS prompting. FS prompting involves including two examples within the prompt: one consisting of a pair of events that we define as having a dependency, and the other consisting of a pair of events for which no dependency is

```
* Background *
(same as ZS template)
* Task description *
(same as ZS template)
* Example of Yes*
Event 1: The campaign of Jill Stein files for a recount in Pennsylvania and plans
to do so in Michigan.
Event 2: Jill Stein files for recount in Michigan.
* Example of No*
Event 1: Saudi Arabia breaks off diplomatic relations with Iran after Sheikh Nimr's
execution and an attack on the Saudi embassy in Tehran.
Event 2: Mexican authorities want to question American actor Sean Penn and
Mexican actress Kate del Castillo about a secret October meeting and interview
with Joaquín Guzmán, published by Rolling Stone magazine yesterday. Mexico
Attorney General Arely Gomez says it was the Penn interview that led authorities
to a Guzman hiding place.
* Event 1:
<event1>
* Event 2:
<event2>
```

Fig. 5. Prompt template for FS

defined. These examples serve as contextual information for the model. To ensure balance, we provide one example each of a positive and a negative case of dependency. Figure 5 presents the template we used for constructing FS prompts. Note that we omit the Background and Task Description sections for clarity in this figure, as their content is identical to that of ZS.

CoT prompting. CoT prompting is a technique that enhances the reasoning capability of LLMs by incorporating intermediate explanations that lead to a final conclusion [27]. We include explanatory reasoning based on whether the two events occurred at the same or nearby locations, or whether they involve shared names of individuals or organizations—factors that can serve as evidence for a dependency as shown in Fig. 6. LLMs are proficient at determining spatial relationships, such as whether two places represent a country-city relation or whether two cities are geographically close. They are also trained on background knowledge about well-known individuals and organizations. Accordingly, our explanations include references to location, time, and, people as potential grounds for inferring dependency. When combining FS and CoT prompting, we used the same examples as those employed in the FS-only setting.

* Example of Yes*
Event 1: The campaign of Jill Stein files for a recount in Pennsylvania and plans to do so in Michigan.
Event 2: Jill Stein files for recount in Michigan.
Reason: These two events relate to the results of an election for the same person. It is also stated in the first event that the second event is scheduled to occur. Therefore, since these events have dependency, the result is Yes.
* Example of No*
Event 1: Saudi Arabia breaks off diplomatic relations with Iran after Sheikh Nimr's execution and an attack on the Saudi embassy in Tehran.
Event 2: Mexican authorities want to question American actor Sean Penn and Mexican actress Kate del Castillo about a secret October meeting and interview with Joaquín Guzmán, published by Rolling Stone magazine yesterday. Mexico Attorney General Arely Gomez says it was the Penn interview that led authorities to a Guzman hiding place.
Reason: These two events illustrate the different relationship between nations. Neither has a common country name. In terms of verbs, the first event is often associated with war, such as "breaks off" or "attack," while the second event is often associated with discussion or negotiation, such as "question" or "led." Because of the difference in subject and action, we can say that there is no dependency between these events, so the answer is no.

Fig. 6. Prompt template for FS-CoT

5 Experimental Evaluation

5.1 Research Questions

We conducted the experiments based on the following research questions (**RQs**):

- **RQ1**: Which LLM model most accurately identified the presence or absence of a dependency between two events?
- **RQ2**: Did the proposed method construct a more effective network compared to previous approaches?

First, we address **RQ1** by evaluating the prediction accuracy of each LLM model using a dataset specifically prepared to verify only the presence or absence of dependencies. Building on these results, we then proceed to **RQ2**, where we identify combinations of events for analyzing dependencies and assess the accuracy of the resulting constructed network.

5.2 Models

We evaluated the following five models to investigate which are most effective for **RQ1**.

- ChatGPT: ChatGPT is based on a transformer architecture and is designed to generate human-like text. The version this study used was GPT-4o.
- Llama 3: This is a transformer-based, autoregressive LLM [26]. We employed the Llama3-8B model in our experiments.
- Mistral: Mistral is a decoder-based model [12]. We used the Mistral-7B-Instruct-v0.1 version, setting the temperature parameter to 0.0001.
- Phi3: Phi3 [1] is a transformer-based model. For our evaluation, we used the Phi-3 Mini model.
- Gemini: Gemini is a multimodal language model [25]. This model is built on a combination of transformers and mixture of experts architecture. Although this model is designed as a multimodal language model, we input only texts. We used the Gemini 2.5 Flash version.

For **RQ2**, we evaluated EEG that investigated whether dependencies between two events could be defined using TF-IDF and cosine similarity to construct event network [30]. EEG determines the presence or absence of a dependency based solely on the textual content of the given sentences.

5.3 Evaluation Criteria

In accordance with prior studies on network construction [30], we employ precision (P), recall (R), and F1 score (F) as defined below.

$$P = \frac{R_A \cap R_M}{R_M}, R = \frac{R_A \cap R_M}{R_A}, F = \frac{2PR}{P + R}$$

where R_A denotes the results obtained by the annotators, whereas R_M indicates the predictions generated by the algorithm.

5.4 Results

RQ1: Which LLM model most accurately identified the presence or absence of a dependency between two events?
A1. Llama 3 achieved the highest F1 score among the models.
A2. Both Llama 3 and ChatGPT achieved F1 scores exceeding 90%.
A3. ZS prompting consistently achieved the highest F1 scores across all LLMs evaluated.
A4. Among the evaluated LLMs, ChatGPT proved to be the most effective for implementing the proposed method.

To appropriately evaluate the performance of each LLM model, we prepared a subset from the dataset created in Section 3. This subset contains two types of data: positive examples, where a dependency exists between two events, and negative examples, where no such dependency exists. Positive examples were directly taken from the existing dataset. Negative examples include randomly

Table 2. Results of dependency prediction by LLM. Bold letters indicate the best results.

	Algorithm	P	R	F
ChatGPT	ZS	84.3%	99.6%	91.3%
	FS	71.7%	**100.0**%	83.5%
	FS-CoT	73.8%	**100.0**%	84.9%
Llama 3	ZS	92.2%	95.1%	**93.6**%
	FS	84.5%	2.5%	4.9%
	FS-CoT	98.9%	4.0%	7.7%
Mistral	ZS	65.3%	99.9%	79.0%
	FS	56.3%	**100.0**%	72.0%
	FS-CoT	56.3%	**100.0**%	72.0%
Phi3	ZS	53.9%	96.3%	69.1%
	FS	53.9%	96.3%	69.1%
	FS-CoT	53.9%	96.3%	69.1%
Gemini	ZS	82.4%	99.8%	90.0%
	FS	74.7%	99.8%	85.5%
	FS-CoT	80.4%	99.5%	88.9%

paired events extracted from different timelines, where human annotators verified that no dependency relationship exists between the events. To eliminate temporal bias, we ensured a balanced selection of events from January 2016 to December 2017. To collect a balanced set of data from many different months, we extracted 100 positive and 100 negative examples per month. However, for July 2017, only 54 positive examples were available; therefore, we also extracted only 54 negative examples for that month. As a result, the subset prepared for evaluating **RQ1** consisted of 2,354 positive examples and 2,354 negative examples, totaling 4,708 event pairs.

Table 2 presents the results of the dependency analysis conducted by all the LLM models used in this study. All LLM models obtained high recall scores. This indicates that they are generally capable of correctly outputting "No" for event pairs that occurred in the same month but were assigned to different timelines by human annotators. On the other hand, precision tends to be lower than recall, suggesting that it is more challenging for these models to correctly output "Yes" for event pairs that should be linked, as well as for human judgment.

Focusing on the F1 scores, Llama 3 achieved the best performance. However, Llama 3 frequently refused to respond to events related to politics or conflicts, often requiring repeated queries to obtain results. Furthermore, Llama 3 did not produce any output for the network selection task. Therefore, in the following experiments, we used ChatGPT, which achieved the second-highest F1 score, for

Table 3. Results of network creation. Bold letters indicate the best results.

Algorithm	P	R	F
EEG	29.8%	26.2%	27.9%
SN-no-opt	41.3 %	16.4%	23.4%
SN-Net	16.1 %	**48.1**%	24.1%
SN-Net-Node	**67.7**%	33.8%	**45.1**%

network construction. In the following, we refer to our method using ChatGPT as SN.

RQ2: Did the proposed method construct a more effective network compared to previous approaches?
A1. We achieved an F1 score approximately 17 percentage points higher than that of previous methods designed for long-form texts.
A2. Node selection in the proposed method made a significant contribution to improving precision and F1 scores.

Table 3 presents the results for EEG; SN-no-opt, which analyzes all combinations of event pairs without applying either network selection or node selection; SN-Net, which applies only the network selection component of our proposed method; and SN-Net-Node, which applies both network selection and node selection. The results show that the proposed method SN-Net-Node achieved the highest F1 score and outperformed the other methods. That is, by selecting in advance both the network to which a newly added event belongs and the set of events within that network, it can be said that even events described in short texts can be used to construct high-accuracy networks when leveraging LLM.

Focusing on SN-Net, we can see that its precision was lower than that of EEG, while its recall was higher. A possible reason could be that SN-Net defined more edges than the human-constructed network, resulting in a denser connection structure. In contrast, by incorporating node selection, SN-Net-Node prevented the network from becoming excessively dense, which in turn led to an improvement in precision. Focusing on precision, SN-Net yielded a lower score than EEG, although SN-no-opt obtained better result than the baseline. This decline can be attributed to the network selection process, which generates multiple event groups and consequently reduces the number of appropriate event pairs analyzed by the LLM. Upon examining the constructed networks, we found that events related to the 2016 U.S. presidential election campaign and those concerning personnel appointments and policy implementations following inauguration that are originally of the new president defined as a single group in the dataset had been divided into two separate groups. As SN-Net generates prompts for all combinations within the same group, this finer-grained division led to denser network constructions than those defined by human annotators. We believe that this over-fragmentation is a key factor contributing to the decline in precision.

Figure 7 illustrates an example of a network generated by SN-Net-Node. In this example, the edges defined by the method are shown as solid lines, while

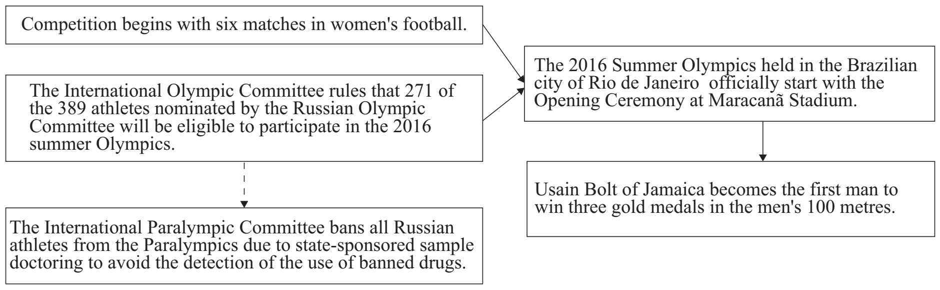

Competition begins with six matches in women's football.
The International Olympic Committee rules that 271 of the 389 athletes nominated by the Russian Olympic Committee will be eligible to participate in the 2016 summer Olympics.
The International Paralympic Committee bans all Russian athletes from the Paralympics due to state-sponsored sample doctoring to avoid the detection of the use of banned drugs.
The 2016 Summer Olympics held in the Brazilian city of Rio de Janeiro officially start with the Opening Ceremony at Maracanã Stadium.
Usain Bolt of Jamaica becomes the first man to win three gold medals in the men's 100 metres.

Fig. 7. An example of an event network created by the proposed method.

the edges that were defined in the dataset but not identified by SN-Net-Node are shown as dashed lines. The solid lines, when viewed in chronological order, successfully trace events related to international sports. However, the method failed to capture connections between events concerning participation of athletes, which were manually defined by humans.

6 Conclusions

Structuring events that are in dependency relationships contributes to understanding the processes that have shaped both past and contemporary societies. In this study, we propose a method to supplement missing contextual information, which cannot be directly analyzed from short texts, by leveraging LLMs, in order to construct networks that represent dependencies among short event descriptions. To achieve this goal, we first created a dataset composed of event networks built from short texts. Using a subset of this dataset, which includes both dependent and independent event pairs, we evaluated the ability of LLMs to appropriately analyze dependencies between short event descriptions. The results demonstrated that Llama 3 and GPT-4o achieved high F1 scores. Subsequently, we proposed a two-step method incorporating network selection and node selection to construct networks while reducing the number of event pairs analyzed by the LLM. We evaluated the effectiveness of this method using the constructed dataset and achieved an F1 score of 45%. This F1 score is approximately 17 percentage points higher than that of previous methods designed for long-form event descriptions, demonstrating the effectiveness of combining LLMs with pair reduction strategies.

As a direction for future work, expanding the dataset used for constructing event networks is a promising avenue. In this study, we focused on data from 2016 and 2017 recorded in current events portal of Wikipedia. However, Wikipedia also contains records of events dating back to ancient times. We will organized all historical records available in Wikipedia by applying our method.

Acknowledgements. This work was supported in part by MEXT Grant-in-Aids (#25K15357). We also thank our anonymous reviewers for their insightful feedback and comments.

References

1. Abdin, M., Aneja, J., et al, H.A.: Phi-3 technical report: A highly capable language model locally on your phone (2024), https://arxiv.org/abs/2404.14219
2. Alonso, O., Shiells, K.: Timelines as summaries of popular scheduled events. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 1037–1044. WWW '13 Companion, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2487788.2488114, https://doi.org/10.1145/2487788.2488114
3. Amayri, O., Bouguila, N.: Online news topic detection and tracking via localized feature selection. In: The 2013 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2013). https://doi.org/10.1109/IJCNN.2013.6707027
4. Brown, T., Mann, B., Ryder, N.e.a.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
5. Chieu, H.L., Lee, Y.K.: Query based event extraction along a timeline. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 425–432. SIGIR '04, Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1008992.1009065, https://doi.org/10.1145/1008992.1009065
6. Duan, Y., Jatowt, A., Yoshikawa, M.: Comparative timeline summarization via dynamic affinity-preserving random walk. In: Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020). Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 1778–1785. IOS Press (2020). https://doi.org/10.3233/FAIA200292, https://doi.org/10.3233/FAIA200292
7. Fan, W., Guo, Z., Bouguila, N., Hou, W.: Clustering-based online news topic detection and tracking through hierarchical bayesian nonparametric models. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2126–2130. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3462982, https://doi.org/10.1145/3404835.3462982
8. Gholipour Ghalandari, D., Ifrim, G.: Examining the state-of-the-art in news timeline summarization. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1322–1334. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.122, https://aclanthology.org/2020.acl-main.122
9. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint arXiv:2203.05794 (2022)
10. Hatzivassiloglou, V., Gravano, L., Maganti, A.: An investigation of linguistic features and clustering algorithms for topical document clustering. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 224–231. SIGIR '00, Association for Computing

Machinery, New York, NY, USA (2000). https://doi.org/10.1145/345508.345582, https://doi.org/10.1145/345508.345582

11. Hayashi, K., Maehara, T., Toyoda, M., Kawarabayashi, K.i.: Real-time top-r topic detection on twitter with topic hijack filtering. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 417–426. KDD '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2783258.2783402, https://doi.org/10.1145/2783258.2783402
12. Jiang, A.Q., Sablayrolles, A., et al, A.M.: Mistral 7b (2023), https://arxiv.org/abs/2310.06825
13. Kurtz, A.J., Mostafa, J.: Topic detection and interest tracking in a dynamic online news source. In: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries. pp. 122–124. JCDL '03, IEEE Computer Society, USA (2003)
14. Liu, B., Han, F.X., Niu, D., Kong, L., Lai, K., Xu, Y.: Story forest: Extracting events and telling stories from breaking news. ACM Trans. Knowl. Discov. Data **14**(3) (May 2020). https://doi.org/10.1145/3377939, https://doi.org/10.1145/3377939
15. Liu, W., Wang, H., Wang, J., Guo, H., Sun, Y., Hou, M., Yu, B., Wang, H., Peng, Q., Zhang, C., Liu, C.: A popular topic detection method based on microblog images and short text information. Journal of Web Semantics **81**, 100820 (2024). https://doi.org/https://doi.org/10.1016/j.websem.2024.100820, https://www.sciencedirect.com/science/article/pii/S1570826824000064
16. Luo, K., Zhou, T., Chen, Y., Zhao, J., Liu, K.: Open event causality extraction by the assistance of LLM in task annotation, dataset, and method. In: Dong, T., Hinrichs, E., Han, Z., Liu, K., Song, Y., Cao, Y., Hempelmann, C.F., Sifa, R. (eds.) Proceedings of the Workshop: Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning (NeusymBridge) @ LREC-COLING-2024. pp. 33–44. ELRA and ICCL, Torino, Italia (May 2024), https://aclanthology.org/2024.neusymbridge-1.4/
17. Mansouri, B., Campos, R., Jatowt, A.: Towards timeline generation with abstract meaning representation. In: Companion Proceedings of the ACM Web Conference 2023. pp. 1204–1207. WWW '23 Companion, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3543873.3587670, https://doi.org/10.1145/3543873.3587670
18. Mao, Q., Li, J., Wang, J., Li, X., Hao, P., Wang, L., Wang, Z.: Explicitly modeling importance and coherence for timeline summarization. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8062–8066 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746383
19. Martschat, S., Markert, K.: A temporally sensitive submodularity framework for timeline summarization. In: Korhonen, A., Titov, I. (eds.) Proceedings of the 22nd Conference on Computational Natural Language Learning. pp. 230–240. Association for Computational Linguistics, Brussels, Belgium (Oct 2018). https://doi.org/10.18653/v1/K18-1023, https://aclanthology.org/K18-1023
20. Qi, Y., Zhou, L., Si, H., Wan, J., Jin, T.: An approach to news event detection and tracking based on stream of online news. In: 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). vol. 2, pp. 193–196 (2017). https://doi.org/10.1109/IHMSC.2017.158
21. Radinsky, K., Horvitz, E.: Mining the web to predict future events. In: Proceedings of the Sixth ACM International Conference on Web Search and Data

Mining. pp. 255–264. WSDM '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2433396.2433431, https://doi.org/10.1145/2433396.2433431

22. Swan, R., Allan, J.: Automatic generation of overview timelines. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 49–56. SIGIR '00, Association for Computing Machinery, New York, NY, USA (2000). https://doi.org/10.1145/345508.345546, https://doi.org/10.1145/345508.345546
23. Takamura, H., Yokono, H., Okumura, M.: Summarizing a document stream. In: Proceedings of the 33rd European Conference on Advances in Information Retrieval. pp. 177–188. ECIR'11, Springer-Verlag, Berlin, Heidelberg (2011)
24. Tan, Z., Zhang, P., Tan, J., Guo, L.: A multi-layer event detection algorithm for detecting global and local hot events in social networks. Procedia Computer Science **29**, 2080–2089 (2014). https://doi.org/https://doi.org/10.1016/j.procs.2014.05.192, https://www.sciencedirect.com/science/article/pii/S187705091400369X, 2014 International Conference on Computational Science
25. Team, G., Anil, R., et al, S.B.: Gemini: A family of highly capable multimodal models (2024), https://arxiv.org/abs/2312.11805
26. Touvron, H., Martin, L., et al, K.S.: Llama 2: Open foundation and fine-tuned chat models (2023), https://arxiv.org/abs/2307.09288
27. Wei, J., Wang, X., Schuurmans, D.e.a.: Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
28. Wu, J., Li, B., Liu, Q.: Topic detection based on bert and seed lda clustering model. In: Proceedings of the 2023 7th International Conference on Innovation in Artificial Intelligence. pp. 72–78. ICIAI '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3594409.3594418, https://doi.org/10.1145/3594409.3594418
29. Xu, G., Meng, Y., Chen, Z., Qiu, X., Wang, C., Yao, H.: Research on topic detection and tracking for online news texts. IEEE Access **7**, 58407–58418 (2019). https://doi.org/10.1109/ACCESS.2019.2914097
30. Yang, C.C., Shi, X., Wei, C.P.: Discovering event evolution graphs from news corpora. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **39**(4), 850–863 (2009). https://doi.org/10.1109/TSMCA.2009.2015885
31. Yu, Y., Jatowt, A., Doucet, A., Sugiyama, K., Yoshikawa, M.: Multi-TimeLine summarization (MTLS): Improving timeline summarization by generating multiple summaries. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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). pp. 377–387. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.32, https://aclanthology.org/2021.acl-long.32
32. Zhang, C., Lyu, J., Xu, K.: A storytree-based model for inter-document causal relation extraction from news articles. Knowl. Inf. Syst. **65**(2), 827–853 (Nov 2022). https://doi.org/10.1007/s10115-022-01781-7, https://doi.org/10.1007/s10115-022-01781-7