Timeline Summarization based on Event Graph Compression via Time-Aware Optimal Transport

Manling Li[1], **Tengfei Ma**[2], **Mo Yu**[2], **Lingfei Wu**[3], **Tian Gao**[2],
Heng Ji[1], **Kathleen McKeown**[4]
[1]University of Illinois Urbana-Champaign
[2]IBM Research [3]JD.COM Silicon Valley Research Center [4]Columbia University
{manling2,hengji}@illinois.edu, tengfei.ma1@ibm.com,
{yum,tgao}@us.ibm.com, lwu@email.wm.edu, kathy@cs.columbia.edu

Abstract

Timeline Summarization identifies major events from a news collection and describes them following temporal order, with key dates tagged. Previous methods generally generate summaries separately for each date after they determine the key dates of events. These methods overlook the events' intra-structures (arguments) and inter-structures (event-event connections). Following a different route, we propose to represent the news articles as an event-graph, thus the summarization task becomes compressing the whole graph to its salient sub-graph. The key hypothesis is that the events connected through shared arguments and temporal order depict the skeleton of a timeline, containing events that are semantically related, structurally salient, and temporally coherent in the global event graph. A *time-aware optimal transport distance* is then introduced for learning the compression model in an unsupervised manner. We show that our approach significantly improves the state of the art on three real-world datasets, including two public standard benchmarks and our newly collected *Timeline*$_{100}$ dataset. [1]

1 Introduction

Timeline summarization (Chieu and Lee, 2004; Yan et al., 2011a,b; Binh Tran et al., 2013; Tran et al., 2013, 2015; Nguyen et al., 2014; Wang et al., 2016; Martschat and Markert, 2018; Steen and Markert, 2019) aims at generating a sequence of major news events with their key dates from a large collection of related news from multiple perspectives (see Figure 1 for an example). The timeline summarization task poses several challenges to existing Natural Language Processing (NLP) techniques: (1) In contrast to multi-document summarization (MDS) dealing with tens of documents (Fabbri et al., 2019), it summarizes hundreds of long documents, which requires the model to efficiently maintain a joint representation of the entire news collection, so that the summary has its coverage and coherence optimized globally. (2) The summary is expected to select key dates and capture the temporal interdependency across key stories, which, compared to standard MDS, poses additional challenges in reconstructing temporal order. (3) Manual labeling of timeline summaries is costly; thus the labeled data for model training is very limited.

As a result, previous studies (Martschat and Markert, 2018; Steen and Markert, 2019) usually take an unsupervised approach. Specifically, these methods first identify the key dates from the publication time distribution. Then for each key date and its associated news articles, a summary is generated based on the salient sentences measured by the inter-similarity of these articles. In these methods, the document representations are limited to local text features, ignoring the global context of the news collection. The applications of neural models, especially advanced pre-trained language models, such as BERT (Devlin et al., 2019a) and GPT-2 (Budzianowski and Vulić, 2019), are restricted in terms of both representation capacity and memory efficiency when handling the global context within such input document size.

We propose an event graph representation along with compression to deal with the representation difficulties in global graph contextualization, scalability, and time-awareness. Our solution consists of the following key ideas.

(1) Event graph construction for multi-doc encoding: With state-of-the-art Information Extraction (IE) systems (Lin et al., 2020), we construct a single event graph from the input documents, with co-referential entities (e.g., *house*, *mansion* in Figure 1) and co-referential events (e.g., *die*, *collapsed*) merged across documents. Our comprehensive event

[1]The programs, data and resources are publicly available for research purpose in https://github.com/limanling/event-graph-summarization.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6443–6456
November 7–11, 2021.

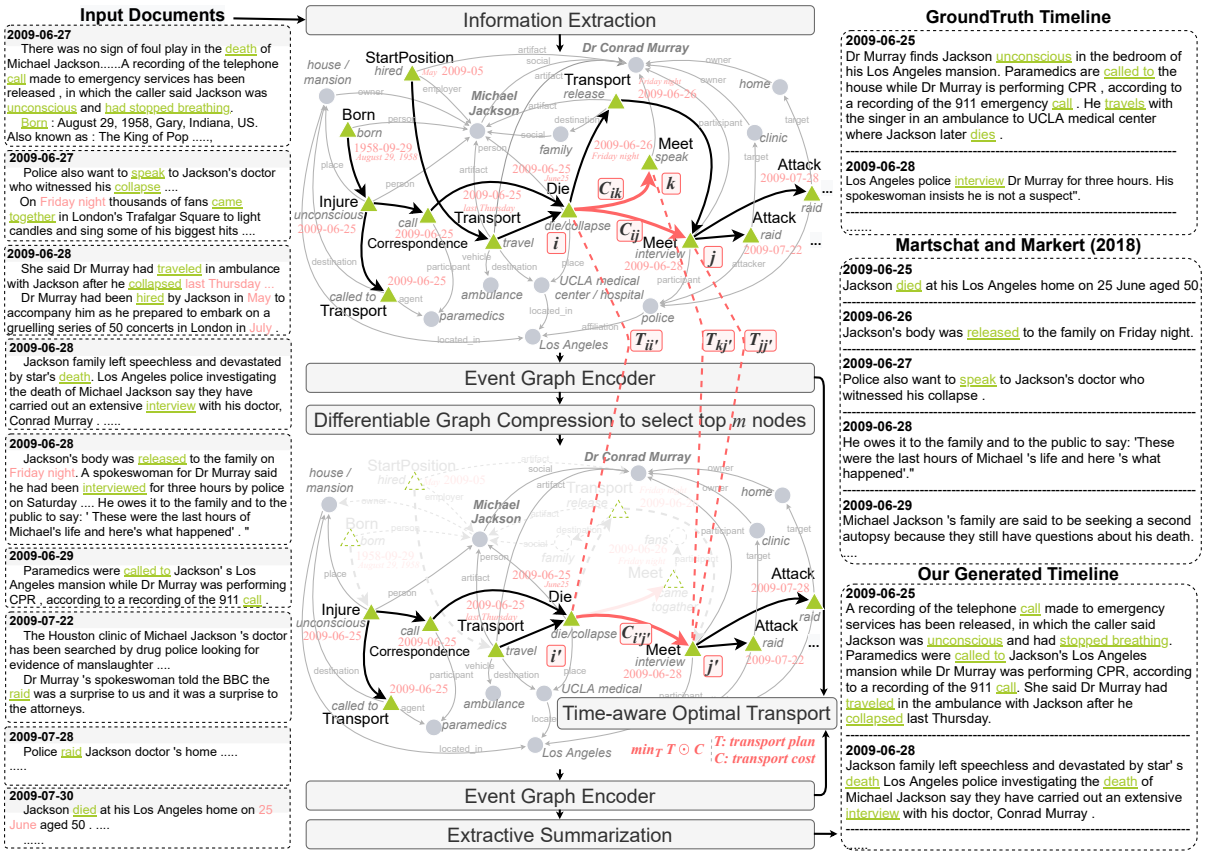

Figure 1: Timeline summarization based on event graph compression. The example is a partial timeline about the investigation on Dr Conrad Murray for the death of Michael Jackson, describing that Michael Jackson is found *unconscious* and Dr Murray *traveled* with him to hospital and started to be *interviewed* by police. We use green triangles to denote events, and grey circles stand for entities. *Italics* represents the raw text mention extracted. Black bold arrows represent the temporal order between events, and grey arrows are event-entity argument edges and entity-entity relation edges. Coreferential events and entities are merged across documents. Faded nodes are events being removed during summarization. In this example, we show the transport of node pairs $\langle i, j \rangle$ and $\langle i, k \rangle$ to the node pair $\langle i', j' \rangle$ in the summary graph.

graph connects events through temporal order (e.g., *interview* $\xrightarrow{\text{BEFORE}}$ *raid*), shared arguments (e.g., *called_to* $\xrightarrow{\text{AGENT}}$ *paramedics* $\xleftarrow{\text{PARTICIPANT}}$ *call*), and related arguments (e.g., *travel* $\xrightarrow{\text{DESTINATION}}$ *hospital* $\xrightarrow{\text{LOCATED\_IN}}$ *Los Angeles* $\xleftarrow{\text{AFFILIATION}}$ *police* $\xleftarrow{\text{PARTICIPANT}}$ *interview*). The graph structure enables the model to capture global long-distance inter-dependency between events across documents.

(2) Unsupervised event graph compression with optimal transport (OT): We propose a new formulation of timeline summarization, by selecting event nodes from the input graph to form a smaller summary graph. Under a certain summary size constraint, a summary graph with high coverage has a small information loss, compared to the one with low coverage (Filatova and Hatzivassiloglou, 2004). We constrain the total number of event nodes to be kept in the summary, and optimize the summary graph to be close to the original graph using optimal transport. The training objective is to find the optimal transport plan between input and summary graph that has the minimal transport distance. Figure 1 shows an example of transporting node pairs in the input graph to the node pair $\langle$*die*, *interview*$\rangle$ in the summary graph. $\langle$*die*, *interview*$\rangle$ receives relatively large mass during the graph transport since it has small distance with multiple node pairs in the input graph, such as $\langle$*die*, *speak*$\rangle$. To obtain the minimal distance with only m events to be kept, a global decision is learned to select salient but also diverse events. The summary graphs are generated using a differentiable compression model according to a hyperparameter of compression rate, instead of using annotated timelines. Thus, our objective allows model training in an end-to-end unsupervised way.

(3) Time-aware Gromov-Wasserstein distance: The distance between two graphs should capture

the following criteria: **i) Semantic relevance:** each node first has its initial *local* context encoded via a pre-trained BERT model and node type embeddings. For example, STARTPOSITION event is not closely related to the TRANSPORT event in Figure 1 though they have temporal dependencies. **ii) Structural centrality:** we employ a graph neural network to maintain a *global* context embedding by encoding the global structure topology, which enables the events of high node centrality to gather comprehensive information from neighbors. For example, although both are MEET events, *interviewed* (by police) is more structurally salient than *speak*. It encodes the information not only from its neighbor events such as *raid*, but also from long-distance neighbors such as *travel* (to hospital) via the aforementioned argument paths. **iii) Temporal coherence:** we define *time-aware Gromov-Wasserstein distance* over the temporal edges, and introduce a *temporal regularizer* to enlarge the distance between events that have wide time gap, such as the BORN and INJURE events in Figure 1, so that the temporal coherence can be captured. It enables the model to select temporally salient events that have temporal dependencies with multiple events in the news collection. Also, timeline summarization is sensitive to temporal ordering, such that the TRANSPORT (*traveling* in ambulance) before DIE in Figure 1 is more important to the story than the TRANSPORT (*releasing* body) after DIE. Hence, we distinguish the before and after events in the distance computation.

(4) New benchmark: Considering the current timeline summarization benchmarks are limited to certain topics, we collect a new dataset *Timeline*$_{100}$ with more testing samples and wider topic coverage. Experiments on three datasets show that our approach is significantly better than the baselines.

2 Method

2.1 Overview

Our approach aims at finding the graph that has minimal distance from the input graph (Filatova and Hatzivassiloglou, 2004), so that when only a limited number of nodes is selected, the summary graph can have menial information loss. Optimal transport is solving this exact problem by finding the best transport plan that has a minimal distance between two graphs. To apply optimal transport to timeline summarization, the key is to design the distance to evaluate the information loss, and thus we propose time-aware optimal transport distance.

Figure 1 gives an overview of our approach. It first extracts an event graph G from input documents. We then encode the graph and perform graph compression to compress G to its summary graph S. Our time-aware optimal transport is applied to train the graph encoder and compression model, with the goal of keeping events that are semantically related, structurally salient, and temporally coherent.

2.2 Event Graph Construction

The event graph is a heterogeneous graph G, where nodes are events $\{v_i\}$ and entities $\{e_j\}$, and edges contain event-event temporal ordering edges $\{\langle v_i, v_l\rangle\}$, event-entity argument edges $\{\langle v_i, a, e_j\rangle\}$, and entity-entity relation edges $\{\langle e_j, r, e_k\rangle\}$. We list all the notations in Table 1.

Symbol	Meaning
G	An input event graph from input documents
n	The number of event nodes in G
S	A summary event graph (a subgraph)
m	The number of event nodes in S
ϕ	A mapping function from a node to its type
w	A mapping function from a node to its mentions
v	An event node in an event graph
e	An entity node in an event graph
$\langle v_i, v_l\rangle$	A temporal ordering edge (v_l happens after v_i)
$\langle v_i, a, e_j\rangle$	An argument edge (the entity e_j plays argument role a in event v_i)
$\langle e_j, r, e_k\rangle$	An entity relation edge between e_j and e_k, and r is the relation type

Table 1: List of symbols.

We apply OneIE (Lin et al., 2020), a state-of-the-art Information Extraction (IE) system, to extract entities, relations and events; then perform cross-document entity and event coreference resolution (Pan et al., 2015, 2017; Lai et al., 2021) over the document cluster of each timeline topic. We apply (Ning et al., 2019) to extract temporal relations for events in the same paragraph or having shared arguments. For example, *clashes* happen before *wound* given the sentence *fifty wounded are reported in the clashes*. To obtain the date of each event, We extract and normalize time expressions using publication date (Manning et al., 2014), and then apply (Wen et al., 2021) to extract the event temporal attributes from the context. If the tem-

poral attributes can not be decided according to the context, we propagate the temporal attributes from neighbor events based on their shared arguments (**?**). After that, we use the document publication date to populate the remaining missing dates. For example, in Figure 1, the date 2009-06-25 of the *collapse* (DIE) event is extracted from context *last Thursday*, and the date of the *unconscious* (INJURE) event is propagated along with their shared argument *Michael Jackson*.

2.3 Time-Aware Optimal Transport (OT)

Optimal Transport. We aim to generate the summary graph S that has minimal OT distance with the input graph G, such that

$$D(G, S) = \min_{\boldsymbol{T}} \boldsymbol{T} \odot \boldsymbol{C},$$

where $\odot$ represents the Hadamard product. $\boldsymbol{T} \in \mathbb{R}_{+}^{n \times m}$ denotes the transport plan, learned to optimize a *soft* node alignment between two graphs. Namely, each node in G can be transferred to multiple nodes in S with different weights. We use $T_{ii'}$ to denote the amount of mass shifted from node i in the input graph G to node i' in the summary graph S, as shown in Figure 1. $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ is the cost matrix of event nodes between two graphs.

Time-Aware OT Distance. Considering that event graphs are heterogeneous graphs, and timeline summarization is sensitive to temporal dependencies between events, we define the Gromov-Wasserstein Distance (Xu et al., 2019) on temporal edges to calculate distance between pairs of nodes within two graphs, i.e., $\langle i, j \rangle$ in G and $\langle i', j' \rangle$ in S:

$$D(G, S) = \min_{\boldsymbol{T}} \sum_{i,j \in G} \sum_{i',j' \in S} T_{ii'} T_{jj'} |C_{ij} - C_{i'j'}|.$$

Figure 1 shows an example of transporting edges $\langle i, j \rangle$ in the input graph to $\langle i', j' \rangle$ in the summary graph. The cost $|C_{ij} - C_{i'j'}|$ evaluates the intra-graph structural similarity between two pairs of nodes $\langle i, j \rangle$ in G and $\langle i', j' \rangle$ in S. To capture the direction of temporal ordering, we parameterize different matrices to distinguish the *before* and *after* nodes:

$$C_{ij} = \| \boldsymbol{W}_{\text{bfr}} \boldsymbol{v}_i - \boldsymbol{W}_{\text{aft}} \boldsymbol{v}_j \|_2 - \Omega(t_i, t_j).$$

In this way, although *travel* in Figure 1 and *release* are both TRANSPORT events connecting with the DIE event, they are distinguished during distance calculation. Here, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the node representations and we want them to capture the semantic relevance, structural salience and temporal coherence. As a result, we design an event graph encoder later in §2.4 from these three aspects.

Temporal Regularizer. The OT distance between events should also capture temporal coherence. For example, in Figure 1, BORN event and INJURE event have large time gap, so that there should be a large distance between them, although they have direct connections in the graph. As a result, we use a regularizer $\Omega(t_i, t_j)$ to penalize events that have a large time difference $t_i - t_j$:

$$\Omega(t_i, t_j) = \frac{\beta}{(t_i - t_j)^2 + 1},$$

where $\beta \in (0, 1]$ is a hyper-parameter.

2.4 Event Graph Encoder

In order to calculate the time-aware optimal transport distance, we encode both the input event graph and the summary graph to obtain the node representations, which capture text semantics, graph structures and preserves the temporal information.

Semantics Encoding. To capture the local text semantics of an entity $\boldsymbol{e}$ or an event $\boldsymbol{v}$, we apply the pre-trained BERT (Devlin et al., 2019b) to initialize a contextualized embedding $\boldsymbol{w}$ using its text mentions. We use the average representation for nodes having multiple mentions, and concatenate it with the node type embedding $\boldsymbol{\phi}$, which is initialized by BERT using the type name. The frequency of events has been proven effective and critical to timeline summarization (Martschat and Markert, 2018). As a result, we add the number of its text mentions $|w|$ to capture the event frequency in the news collection:

$$\boldsymbol{v} = [\boldsymbol{w}_v; \boldsymbol{\phi}_v; |w_v|], \boldsymbol{e} = [\boldsymbol{w}_e; \boldsymbol{\phi}_e; |w_e|], \quad (1)$$

where $[;]$ denotes concatenation operation.

Graph Encoding. After that, we employ an edge-wise graph neural network to contextualize all the nodes with their global graph contexts. We first generate edge type representation $\boldsymbol{a}$ and $\boldsymbol{r}$ by encoding the edge type name using pre-trained BERT, and temporal edge representation $\boldsymbol{t}$ is encoded using name "before". The message passed through an argument edge $\langle v_i, r, e_j \rangle$ is:

$$\boldsymbol{m}_{i,j} = \text{ReLU}\left[(\boldsymbol{W}_a [(\boldsymbol{v}_i - \boldsymbol{e}_j); \boldsymbol{a}]\right).$$

The messages of relation and temporal edges are similar, by replacing $\boldsymbol{a}$ with $\boldsymbol{r}$ and $\boldsymbol{t}$. We aggregate

the messages using edge-aware attention following (Liao et al., 2019),

$$\alpha_{i,j} = \sigma(\text{MLP}(\boldsymbol{v}_i - \boldsymbol{v_j})),$$

where σ denotes sigmoid function. We adopt a two-layer MLP with ReLU as activation funtion.

The event node representation $\boldsymbol{v}_i$ is then updated using the messages from its local neighbors $N(v_i)$:

$$\boldsymbol{v}_i \leftarrow \text{GRU}\left(\left[\boldsymbol{v}_i; \sum\nolimits_{j \in N(v_i)} \alpha_{i,j} \boldsymbol{m}_{i,j}\right]\right),$$

similar to entity node representations.

Date Distribution Encoding. To encode the date distribution, for each event v_i with date t_i, we concatenate the above node representation $\boldsymbol{v}_i$ with the number of documents published on t_i, the number of events happening on t_i, and the number of event text mentions attached to t_i in local context. It enables the OT distance to capture the corpus-level date salience.

2.5 Differentiable Graph Compression

To get a summary graph with m event nodes [2], we apply an event graph compression matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ following (Ma and Chen, 2021),

$$\boldsymbol{A}_S = \boldsymbol{M}^T \boldsymbol{A}_G \boldsymbol{M},$$

where $\boldsymbol{A}_G \in \mathbb{R}^{n \times n}$ is the temporal edge adjacency matrix of event nodes in G, with $\boldsymbol{A}_S \in \mathbb{R}^{m \times m}$ for S similarly. For timeline summarization task, the parametrization of $\boldsymbol{M}$ has two requirements: (1) $\boldsymbol{M}$ is differentiable to enable end-to-end training; (2) we want to guarantee that the nodes in the summary graph are originally from the input graph (due to our extractive summarization goal), so we follow (Ma and Chen, 2021) to directly select nodes as summary nodes according to their weights $\boldsymbol{\alpha} \in \mathbb{R}^{n \times 1}$:

$$\boldsymbol{\alpha} = \sigma(\widehat{\boldsymbol{A}} \boldsymbol{V} \boldsymbol{W}_\alpha)$$

Here, $\widehat{\boldsymbol{A}} \in \mathbb{R}^{n \times n}$ is the normalized graph adjacency matrix defined in graph convolutional networks (Kipf and Welling, 2017), $\boldsymbol{V} \in \mathbb{R}^{n \times d}$ is the node feature matrix, and $\boldsymbol{W}_\alpha \in \mathbb{R}^{d \times 1}$ is a parameter vector. σ is the sigmoid function.

We pick the top m values of $\boldsymbol{\alpha}$ and list them in the sorted order, denoted by $\boldsymbol{\alpha}_s \in \mathbb{R}^{m \times 1}$. Similarly, $\widehat{\boldsymbol{A}}_s \in \mathbb{R}^{n \times m}$ is the column-sorted and picked version of $\widehat{\boldsymbol{A}}$. Then the compression matrix $\boldsymbol{M}$ can be finally defined as

$$\boldsymbol{M} = \ell_1\text{-row-normalize}[\widehat{\boldsymbol{A}}_s \odot (\boldsymbol{1}\boldsymbol{\alpha}_s^T)],$$

where $\boldsymbol{1}$ means a column vector of all ones.

2.6 Training Objective

The optimal $\boldsymbol{T}$ that solves $D(G, S) = \min_{\boldsymbol{T}} \boldsymbol{T} \odot \boldsymbol{C}$ can be approximated by a differentiable Sinkhorn-Knopp algorithm (Sinkhorn, 1964; Cuturi, 2013) following (Xu et al., 2019; Ma and Chen, 2021),

$$\boldsymbol{T} = \text{diag}(\boldsymbol{p}) \exp(-\boldsymbol{C}/\gamma) \text{ diag}(\boldsymbol{q}),$$

where $\boldsymbol{p} \in \mathbb{R}_+^{n \times 1}$ and $\boldsymbol{q} \in \mathbb{R}_+^{m \times 1}$. The solution $\boldsymbol{T}$ can be computationally obtained by using Sinkhorn's algorithm. Starting with any positive vector $\boldsymbol{q}^0$ to perform the following iteration:

$$\begin{aligned} &\text{for } \ i = 0, 1, 2, \ldots \text{ until convergence,} \\ &\quad \boldsymbol{p}^{i+1} = \boldsymbol{1} \oslash (\boldsymbol{K}\boldsymbol{q}^i), \\ &\quad \boldsymbol{q}^{i+1} = \boldsymbol{1} \oslash (\boldsymbol{K}^\top \boldsymbol{p}^{i+1}), \end{aligned}$$

where $\oslash$ denotes element-wise division. A computational $\boldsymbol{T}^k$ can be obtained by iterating a finite number k times,

$$\boldsymbol{T}^k := \text{diag}(\boldsymbol{p}^k) \boldsymbol{K} \text{diag}(\boldsymbol{q}^k).$$

The parameterization of the graph compression step and Sinkhorn-Knopp algorithm are differentiable, so we can optimize our time-aware optimal transport distance between two graphs in an end-to-end manner.

The advantage of our approach is that the training process is unsupervised, since the summary graph is generated automatically under the constraint of the hyperparameter m, i.e., the number of event nodes in the summary graph. The model parameters include those for the *graph encoder* (capturing semantic relevance, structural centrality and time salience), the *transport distance matrix* (capturing temporal coherence), the *compression model* (selecting top ranked nodes in a differentiable manner), and the *transport plan* (making a global decision to obtain minimum distance). They are optimized jointly to minimize the distance between the generated graph and the input graph.

2.7 Extractive Summarization

During summarization, the event summary graph is generated by selecting m events according to

[2] We only compress the event nodes since that the key for timeline summarization is salient event selection, while arguments are used to capture the distance between events.

the event weights $\boldsymbol{\alpha}$, where m is a hyperparameter decided by the expected compression rate. To maintain the diversity of the temporal dimension following (Martschat and Markert, 2018), we set a maximum event constraint to select no more than k events for each date. In detail, if the event number of one date reaches the limitation, the remaining events of that date will be ignored in the ranking list $\boldsymbol{\alpha}$, and only events happening on other dates can be selected to the summary graph. For each date, k is decided by the date distribution (i.e., the number of events happening on each date), as well as the compression rate hyperparameter.

Finally, for each event $v \in V_S$ in the summary graph, we extract an event summary sentence, i.e., the source sentence with the maximum event coverage.[3] The event summaries are ordered by dates to form the timeline. The event summaries on the same date are merged following the events' temporal orders with topological sort (Manber, 1989).

3 Experiment

3.1 Experimental Settings

Datasets. The evaluation is conducted on three datasets. *Timeline$_{17}$* (Tran et al., 2013) and *Crisis* (Tran et al., 2015) are two widely used timeline summarization datasets. *Timeline$_{17}$* contains 17 topics, and each topic has 1-3 ground-truth timelines, resulting in 19 timelines in total. *Crisis* has 5 topics and each topic has 4-7 ground-truth timelines, with 22 timelines annotated in total. We use all 19 and 22 timelines as references, and calculate the average scores following previous work.

To explore the robustness of our event graph compression for different scenarios, we also collect a new larger dataset *Timeline$_{100}$* containing 100 timelines from news websites including VoA[4] and Reuters[5]. The timelines are written by journalists and are manually curated. The dataset covers various topics related to the *economy*, *military*, *education*, etc. The input documents for each timeline are selected using BM25 (Robertson et al., 1995). For each dataset, we construct input event graphs following §2.2. [6] We use the ACE event ontology[7], with 7 entity types, 6 relation types, 33 event types, and 22 argument roles. For the (unsupervised) training of our event graph compression model, we use event graphs constructed from VoA news between 2011 and 2017 (Li et al., 2020a). The statistics are shown in Table 2.

Dataset	Split	#Doc	#Event	#Entity	#Rel
Timeline$_{17}$	Input	4,650	74,320	115,585	136,509
	Timeline	19	974	1,936	1,134
Crisis	Input	20,463	325,695	551,228	610,410
	Timeline	22	736	1,184	1,309
Timeline$_{100}$	Input	10,379	178,581	301,132	306,975
	Timeline	100	3,296	8,901	23,732
Unlabeled (for OT)	Input	72,576	913,679	381,735	1,046,066
	Timeline	-	-	-	-

Table 2: Data statistics, including the number of documents, events, entities, and temporal relations.

Evaluation Metrics. We use the conventional metrics for timeline summarization (Martschat and Markert, 2018) to evaluate the key date selection using *Date F_1* and the content generation using ROUGE scores, including (1) *concat F_1* to compute ROUGE by concatenating the summaries of all selected dates; (2) *agree F_1* to compute ROUGE only between the summaries which have the same dates; (3) *align F_1* to first align summaries in the output with those in the reference based on similarity and the distance between their dates, then compute the ROUGE score between aligned summaries. Distant alignments are punished.

Baselines. We compare with: (1) (Chieu and Lee, 2004), a typical extractive model based on sentence similarity; and (2) (Martschat and Markert, 2018), the state-of-the-art extractive timeline sumarization model based on submodular functions. (3) PacSum (Zheng and Lapata, 2019), the state-of-the-art unsupervised graph-based ranking summarization baseline, which utilizes BERT to encode sentences for sentence centrality ranking in a sentence graph. We use the publication date of the selected sentence as key dates. (4) SummPip (Zhao et al., 2020), the state-of-the-art unsupervised multi-document summarization baseline, which constructs a sentence graph and performs spectral clustering. After that, a summary is generated for each sentence cluster

[3] We select the events with highest temporal attribute accuracy if there is a tie. The events with temporal attributes extracted directly from the context are of highest priority, followed by events having temporal attributes propagated from neighbor events in §2.2, and then the ones using document publication date.

[4] https://wwconw.voanews.com

[5] https://www.reuters.com

[6] The preprocessed event graphs are released together with the dataset.

[7] https://www.ldc.upenn.edu/collaborations/past-projects/ace

Dataset	Model	Concat F_1 R-1	Concat F_1 R-2	Agree F_1 R-1	Agree F_1 R-2	Align F_1 R-1	Align F_1 R-2	Date F_1
Timeline_{17}	Chieu and Lee (2004)	0.223	0.049	0.024	0.008	0.046	0.012	0.195
	Martschat and Markert (2018)	0.364	0.087	0.092	0.021	0.103	0.024	0.543
	Zheng and Lapata (2019)	0.231	0.054	0.029	0.012	0.035	0.013	0.173
	Zhao et al. (2020)	0.242	0.057	0.028	0.009	0.030	0.007	0.158
	Optimal Transport	0.370	0.089	0.092	0.020	0.103	0.024	0.550
	w/o temporal regularizer	0.369	0.087	0.091	0.018	0.101	0.025	0.545
Crisis	Chieu and Lee (2004)	0.348	0.065	0.026	0.006	0.047	0.010	0.146
	Martschat and Markert (2018)	0.333	0.071	0.056	0.012	0.076	0.015	0.288
	Zheng and Lapata (2019)	0.144	0.017	0.004	0.001	0.008	0.001	0.077
	Zhao et al. (2020)	0.124	0.016	0.004	0.001	0.007	0.001	0.069
	Optimal Transport	0.348	0.074	0.058	0.012	0.079	0.015	0.291
	w/o temporal regularizer	0.348	0.073	0.056	0.011	0.076	0.014	0.290
Timeline_{100}	Chieu and Lee (2004)	0.127	0.028	0.011	0.003	0.017	0.004	0.138
	Martschat and Markert (2018)	0.257	0.060	0.016	0.005	0.021	0.007	0.290
	Zheng and Lapata (2019)	0.219	0.045	0.011	0.002	0.016	0.005	0.151
	Zhao et al. (2020)	0.196	0.034	0.011	0.002	0.017	0.004	0.158
	Optimal Transport	0.278	0.067	0.017	0.005	0.023	0.008	0.295
	w/o temporal regularizer	0.279	0.067	0.015	0.004	0.021	0.007	0.292

Table 3: Performance on timeline summarization. R-1 and R-2 represents ROUGE-1 and ROUGE-2, respectively.

by multi-sentence compression, and we use the most frequent publication date of the sentences in the cluster as key dates. (5) "w/o temporal regularizer", an ablation study by removing the temporal regularizer in the OT distance. [8]

Training Details. The dimension of contextual embedding, type embedding, and edge embedding are 768. β is 0.5. γ is 1. The ratio of event nodes kept after compression m is determined based on the ratio of input graph size and summary graph size of the dataset. We use 0.05 for *Timeline*$_{17}$ dataset, 0.005 for *Crisis* dataset, and 0.05 for *Timeline*$_{100}$ dataset [9]. Due to the large size of input graphs, we first compress the subgraph extracted from each publication date following the hard cutoff of (Martschat and Markert, 2018), and then compress the graph of the entire corpus. The graph compression model is trained on one Tesla V100 GPU with 16GB DRAM.

3.2 Quantitative Performance

As shown in Table 3, our method outperforms baselines on all three datasets. Event graph connects events through entities and temporal relations, which enables capturing the correspondence between events, and excludes unrelated events. General multi-document summarization and text graph based summarization cannot capture the temporal dimension, so the performance is especially low on date F_1, agree F_1 and align F_1. All Concat F_1 scores are significantly different from baselines with p value less than 0.05.

Removing the temporal regularizer results in a consistent performance drop on date F1, showing that our time-aware OT helps select events that are temporally coherent.

We achieve larger gains compared to baselines on *Crisis* dataset, which has larger input graph size and compression rate according to Table 2. It proves the effectiveness of our event graph on encoding a large number of documents and perform effective summarization. Compared to Timeline_{17}, the performance gain on Timeline_{100} is larger, which cover more scenarios. It demonstrates the robustness of our event graph compression method.

3.3 Qualitative Analysis

Figure 1 shows an example of generated timeline comparing with the reference timeline and the best performing baseline (Martschat and Markert, 2018). The number of dates selected by the baseline is larger compared to our approach, which demonstrates that our approach can better detect salience of dates. We think this is because we take advantage of event graphs to capture the events that are temporally salient. For example, our approach avoids the dates that do not have associated salient

[8] For fair comparison, our baselines focus on unsupervised methods that can produce key dates, which excludes text word graph based models and pretrained langauage model based generation models due to lack of temporal dimensions.

[9] We choose m based on three times of reference compression rates to allow comprehensive information being kept.

events, such as 2009-06-26. Also, our temporal attributes are more comprehensive and accurate due to the attribute propagation through shared arguments. For example, the dates of *unconscious* and *travel* in Figure 1 are propagated from the *die* event via the shared argument *Michael Jackson*.

Compared to the baselines, our approach keeps more events in the summary (highlighted in green in Figure 1), while the baseline may produce a summary without events included, e.g., the summary of 2009-06-29.

Compared to the reference timeline, our model is shown to successfully detect the salient events in the graph compression process. Although the *release* event has connections to multiple events, it is not semantically relevant to other events, and thus it will not receive a large mass during the transportation. The *speak* event is not strongly connected to other nodes, and it is semantically close to *interview*, which will not be selected in the global decision of the optimal transport plan. Similarly, the *born* event is omitted due to its large time gap with other events, and the *hire* event is excluded since it is not semantically related to other events. More examples are included in the Appendix.

3.4 Human Evaluation

We follow previous work (Steen and Markert, 2019) to do a scoring-based evaluation. We instruct the human annotators to read 15 randomly sampled reference timelines, and rate summaries generated by our system and baselines on a 1-5 point scale (1 is the worst and 5 is the best). We provide reference timelines as the gold standard to annotators, instead of providing the input news collection. It is because that each timeline contains hundreds of long documents as input, making it hard to judge coverage and control scoring standards of multiple annotators. As the evaluation is scoring-based, we only ask one annotator to score all timelines of each topic to guarantee the same scoring standard. The order of annotating timelines is random, and the annotators have no knowledge about the order of the systems. Each timeline annotation takes around thirty minutes.

The timelines are evaluated in the following dimensions: (1) *general score*: the general quality of the timeline; (2) *coverage score*: the events that are covered by the timeline; (3) *coherence score*: the coherence of the story; (4) *temporal preserving score*: the selection of key dates. Table 4 shows that our approach gets better results on all four measures, proving that our model is reasonable to find semantically relevant, structurally salient and temporally coherent events.

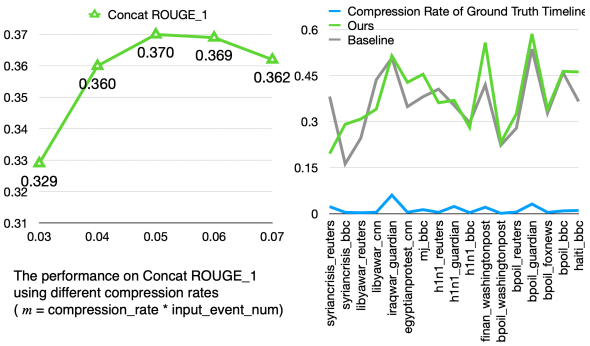

Figure 2: Analysis about compression rates.

3.5 Discussions

Generation Length. Previous work on timeline summarization (Chieu and Lee, 2004; Martschat and Markert, 2018) relies on the reference timeline to decide the compression parameters, such as the overall length or the number of days. In our model, the number of nodes to be kept is decided by the hyperparameter m. Following previous work, we choose m based on the reference compression rate, i.e., the ratio of the event nodes in reference summary to the input event nodes, as detailed in §3.1. Figure 2 shows the relevance between the performance and compression rate.

Compression Rate. The summarization performance is affected by the compression rate of the reference summary. Figure 2 shows that our model achieves larger gains compared to baselines on the timeline with higher reference compression rates, demonstrating that our model is able to effectively select salient events for a large input corpus.

Timeline Topics. Figure 2 shows that the compression rates do not have correlations with timeline topics, and our performance gains compared to baselines are not closely related to timeline topics, proving the robustness of our method.

Input Graph Size. When generating timelines for the same complex event *BP Oil Spill*, as shown in Table 5, the performance gain is generally increasing with respect to the input graph size. It proves the effectiveness of our model on selecting salient information from large graphs.

Model	General	Coverage	Coherence	Temporal Preserving
Chieu and Lee (2004)	2.4	1.4	2.5	1.4
Martschat and Markert (2018)	3.2	2.7	3.4	2.6
Optimal Transport	3.9	2.9	3.7	2.8

Table 4: Human evaluation on a scale of 1-5 (1 is the worst and 5 is the best).

Topic: BP Oil Spill	Graph Size	Concat ROUGE_1 Ours	Baseline	Δ
bpoil_washingtonpost	2582	0.232	0.223	+0.009
bpoil_guardian	2744	0.566	0.536	+0.029
bpoil_bbc	2972	0.464	0.459	+0.004
bpoil_foxnews	3032	0.341	0.328	+0.014
bpoil_reuters	3488	0.326	0.279	+0.047

Table 5: Analysis on the size of input event graph.

4 Related Work

Multi-Document Summarization. Graph-based MDS methods (Barzilay et al., 1999; Erkan and Radev, 2004; Haghighi and Vanderwende, 2009; Ganesan et al., 2010; Banerjee et al., 2015; Yasunaga et al., 2017; Fabbri et al., 2019; Liu and Lapata, 2019; Wang et al., 2020; Huang et al., 2020) are closely related to timeline summarization but cannot be directly applied, due to the lack of temporal dimensions.

Timeline Summarization. Due to the lack of training data, timeline summarization focuses on extractive methods with heuristics (Chieu and Lee, 2004; Yan et al., 2011a,b; Binh Tran et al., 2013; Tran et al., 2013, 2015; Nguyen et al., 2014; Wang et al., 2016; Martschat and Markert, 2018), with a few abstractive methods (Steen and Markert, 2019; Chen et al., 2019; Ansah et al., 2019) that require a few gold summaries to work. They both fail to capture the rich event structures and ignore the temporal orders between events. We are the first to use optimal transport on summarization task to select semantic relevant, structurally salient and temporally coherent events.

Graph Representation of Documents. In general NLP research, people have built various text graphs by augmenting original text sequences with different hidden structural information, such as entity-centric graphs for efficient joint-encoding of large corpora (Wu et al., 2021; De Cao et al., 2019; Ding et al., 2019; Asai et al., 2020; Min et al., 2019; Das et al., 2019). Event graphs from a single document have been built for event schema induction (Li et al., 2018, 2020b), event coreference resolution (Phung et al., 2021; Zeng et al., 2021), etc. However, they ignore relations between event arguments, or only use hierarchical or temporal relations to connect events. Also, cross-document entity coreference and event coreference resolution are critical for large corpora understanding, while previous work focuses on a single document. Our approach is unique in building event-centric graphs across documents, with rich argument and temporal information.

5 Conclusions and Future Work

We propose a novel event graph compression framework for timeline summarization and achieve state-of-the-art on multiple real-world datasets. Our usage of event graphs allows for efficient joint encoding of a large number of documents; and our proposed time-aware optimal transport allows unsupervised training of the entire framework. Future work includes extending our approach to abstractive summarization, and adding subevent relation to hierarchically generate the timeline.

Acknowledgement

This research is based upon work in part supported by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, FA8750-19-C-0206, the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract No. FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Jeffery Ansah, Lin Liu, Wei Kang, Selasi Kwashie, Jixue Li, and Jiuyong Li. 2019. A graph is worth a thousand words: Telling event stories using timeline summarization graphs. In *The World Wide Web*

Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pages 2565–2571. ACM.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ILP based multi-sentence compression. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1208–1214. AAAI Press.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA. Association for Computational Linguistics.

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*, pages 91–92.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 2019. Learning towards abstractive timeline summarization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4939–4945. ijcai.org.

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*, pages 425–432.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Text Summarization Branches Out*, pages 104–111, Barcelona, Spain. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado. Association for Computational Linguistics.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Tuan Lai, Heng Ji, Trung Bui, Quan Hung Tran, Franck Dernoncourt, and Walter Chang. 2021. A context-dependent gated module for incorporating symbolic semantics into event coreference resolution. *arXiv preprint arXiv:2104.01697*.

Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020a. Cross-media structured common space for multimedia event extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2557–2568, Online. Association for Computational Linguistics.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020b. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, Online. Association for Computational Linguistics.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4201–4207. ijcai.org.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. 2019. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4257–4267.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Tengfei Ma and Jie Chen. 2021. Unsupervised learning of graph hierarchical abstractions with differentiable coarsening and optimal transport. *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Udi Manber. 1989. *Introduction to algorithms: a creative approach*. Addison-Wesley Longman Publishing Co., Inc.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Sebastian Martschat and Katja Markert. 2018. A temporally sensitive submodularity framework for timeline summarization. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 230–240, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Kiem-Hieu Nguyen, Xavier Tannier, and Veronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1208–1217, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on*

Natural Language Processing (EMNLP-IJCNLP), pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with Abstract Meaning Representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1130–1139, Denver, Colorado. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Duy Phung, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2021. Hierarchical graph convolutional networks for jointly resolving cross-document coreference of entity and event mentions. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 32–41.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Richard Sinkhorn. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31, Hong Kong, China. Association for Computational Linguistics.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *European Conference on Information Retrieval*, pages 245–256. Springer.

Giang Binh Tran, Tuan A Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. Leveraging learning to rank in an optimization framework for timeline summarization. In *SIGIR 2013 Workshop on Time-aware Information Access (TAIA)*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. 2016. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 58–68, San Diego, California. Association for Computational Linguistics.

Haoyang Wen, Yanru Qu, Heng Ji, Qiang Ning, Jiawei Han, Avirup Sil, Hanghang Tong, and Dan Roth. 2021. Event time extraction and propagation via graph attention networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 62–73.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2021. Graph neural networks for natural language processing: A survey. *arXiv preprint arXiv:2106.06090*.

Hongteng Xu, Dixin Luo, and Lawrence Carin. 2019. Scalable gromov-wasserstein learning for graph partitioning and matching. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3046–3056.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011a. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 433–443, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011b. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 745–754. ACM.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Qi Zeng, Manling Li, Tuan Lai, Heng Ji, Mohit Bansal, and Hanghang Tong. 2021. Gene: Global event network embedding. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 42–53.

Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza

Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1949–1952. ACM.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Example Output

Method	Example Output
Reference	**2011-02-18** Libyan state television shows images of men chanting pro-Gadhafi slogans , waving flags and singing around the Libyan leader 's limousine as it creeps through Tripoli . In Benghazi , human rights groups and protesters claim they 're under attack by pro-government security forces . Among the tens of thousands of protesters who take to the streets , at least 20 people are killed and 200 are wounded , according to medical sources . **2011-02-19** Protests continue to turn violent , however the death and injury toll is unclear . In Benghazi , witnesses report bloody clashes with soldiers firing tear gas and bullets . Witnesses say protests have erupted in cities across the country . Human Rights Watch reports that 84 people have been killed in Libyan demonstrations since February 15 . **2011-02-20** Violence surges in Benghazi where a witness says protesters have taken control of the city and much of Tripoli . Gadhafi 's son Saif al-Islam Gadhafi appears on state television to warn demonstrators that the country could fall into civil war if their protests do not subside . **2011-02-21** The Libyan newspaper Quryna reports that the country 's justice minister has resigned to protest what he calls a " bloody situation and use of excessive force " by security forces against protesters.
Chieu and Lee (2004)	**2011-02-21** By the CNN Wire Staff Libya protests spread to Tripoli State Department has ordered the evacuation of all non-essential personnel The Obama administration is stressing the need to avoid violence against protesters Gadhafi 's son has warned of a possible civil war if protesters do not back down Washington (CNN) – The United States on Monday condemned the violence in Libya and called for a halt to the " unacceptable bloodshed " in response to civil unrest , Secretary of State Hillary Clinton said in a statement .
Martschat and Markert (2018)	**2011-02-15** Protests began February 15 in the eastern city of Benghazi , Libya 's second largest . Witness says square in Benghazi is full of protesters , but there is little sign of police or military Tanks surrounded demonstrators in Benghazi , a protester says 50 reportedly killed since Tuesday , 20 of them Friday U.S. president condemns the government crackdowns in Libya , Bahrain and Yemen (CNN) – At least 20 people were killed and 200 more were injured Friday in the northern Mediterranean city of Benghazi , Libya 's second-largest , said a medical source in Benghazi who was not identified for security reasons . **2011-02-21** Among other things , Washington was taking a close look at a speech early Monday by Saif al-Islam Gadhafi – the Libyan leader 's son – which included warnings of a civil war if demonstrations in the North African country do n't stop . The United States on Monday condemned the violence in Libya and called for a halt to the " unacceptable bloodshed " in response to civil unrest , Secretary of State Hillary Clinton said in a statement .
Ours	**2011-02-16** Source : Several people arrested after police confronted protesters in Benghazi , Libya . **2011-02-18** An Iranian opposition member warns that street protests could lead to civil war " Nastaran " warns that protests are strengthening Iran ' s Revolutionary Guard and pro - government militia. **2011-02-19** A Libyan woman supportive of the protesters , who was not identified to protect her safety , told CNN that army soldiers on Saturday initially claimed solidarity with the demonstrators , only to reverse their tack and open fire on the crowd . Three of those injured are in critical condition , the sources said . While Human Rights Watch , citing interviews with hospital staff and witnesses , reported 84 deaths since Tuesday , the total number is unknown and could n ' t be independently confirmed by CNN . **2011-02-20** Protests continue to turn violent , however the death and injury toll is unclear .

Table 6: Example Output. The event triggers are highlighted in red. The date seletion and event node coverage of our method are much higher compared to baselines.