Joint Learning-based Heterogeneous Graph Attention Network for Timeline Summarization

Jingyi You, Dongyuan Li, Hidetaka Kamigaito,
Kotaro Funakoshi and Manabu Okumura
Tokyo Institute of Technology
{youjy,lidy,kamigaito,funakoshi,oku}@lr.pi.titech.ac.jp

Abstract

Previous studies on the timeline summarization (TLS) task ignored the information interaction between sentences and dates, and adopted pre-defined unlearnable representations for them. They also considered date selection and event detection as two independent tasks, which makes it impossible to integrate their advantages and obtain a globally optimal summary. In this paper, we present a *joint learning-based heterogeneous graph attention network for TLS* (HeterTLS), in which date selection and event detection are combined into a unified framework to improve the extraction accuracy and remove redundant sentences simultaneously. Our heterogeneous graph involves multiple types of nodes, the representations of which are iteratively learned across the heterogeneous graph attention layer. We evaluated our model on four datasets, and found that it significantly outperformed the current state-of-the-art baselines with regard to ROUGE scores and date selection metrics.

1 Introduction

Timeline summarization (TLS) is designed to extract sentences that describe an evolutionary story from a massive amount of web articles with respect to a specific topic in chronological order. TLS has drawn much attention in recent years (Chen et al., 2019; Martschat and Markert, 2018; You et al., 2021b; Ghalandari and Ifrim, 2020; Yu et al., 2021) since it releases people from burdensome manual creation of summaries and gives readers a faster but comprehensive access to track events from many aspects, such as start and end, causality, and the main protagonists involved.

Most studies on TLS seek ways to combine two individual subtasks: date selection and event detection. Depending on different strategies for them, current methods are generally divided into three categories (Ghalandari and Ifrim, 2020): 1) *direct summarization* approaches (Chieu and Lee, 2004; Tran et al., 2013; Martschat and Markert, 2018; Duan et al., 2020) directly identify topic-related sentences from a collection of news articles to form a timeline; 2) *date-wise summarization* methods (Wang et al., 2016; Ghalandari and Ifrim, 2020; Li et al., 2021; Quatra et al., 2021) first select salient dates then construct a timeline for each date individually with sentences of the highest score; and 3) *event detection* algorithms (Steen and Markert, 2019; Duan et al., 2020; Yu et al., 2021) detect events by clustering sentences from multi-timeline news articles then identify several of the most important events and summarize them separately.

Although great successes have been achieved in conducting TLS, several issues remain unsolved. First, current TLS methods mainly adopt statistical hand-designed features to represent dates, e.g., the number of published articles and topic-related sentences in a specific time duration (Yu et al., 2021; Ghalandari and Ifrim, 2020), and employ sentence-BERT (Reimers and Gurevych, 2019) and other pre-defined unchangeable representations for sentences. The low-level or unlearnable representations tend to ignore the semantic and temporal information interaction between sentences and dates, which significantly degrades the performance of downstream tasks. Secondly, traditional approaches focus on either date selection or event detection. Although excellent date selection algorithms can pinpoint accurate timeline dates, they usually extract topic-irrelevant sentences. While event detection algorithms are capable of avoiding redundant summaries by various clustering strategies, they sometimes capture wrong timeline dates. To the best of our knowledge, there is no framework that jointly learns the advantages of the above two subtasks to accurately capture salient dates and eliminate topic-irrelevant sentences in a timeline.

To circumvent the above dilemma, we propose to jointly learn date/sentence representations and event detection-based sentence clustering in a het-

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4091 - 4104

erogeneous graph attention network (HAN) for TLS. Specifically, we construct a heterogeneous graph with dates, words, and sentences as semantic units to solve the first problem. In this graph, words act as a bridge between dates and sentences, enabling date nodes to learn different granularities (word- and sentence-level) of semantic information and sentence representations to be complemented with a date-related intra- and cross-sentence message. As for the second issue, semi-supervised date prediction and event detection-based clustering are integrated into an overall objective, where labeled dates guide and facilitate sentence clustering, and sentence-level clustering information indicating main events improves the accuracy of unlabeled date prediction. Note that we create a new way beyond the above-mentioned three categories as a *joint end-to-end* approach since we no longer have to handle each subtask step by step.

We highlight our contributions as follows:

- This study is the first to construct a model for automatic TLS as a HAN that propagates heterogeneous information with different granularities, of *date-word-sentence*, to effectively learn flexible and accurate representations for both date and sentence nodes.
- Date selection and event detection subtasks are incorporated into an overall objective so that they can be jointly optimized to obtain a globally optimal solution.
- We have empirically shown that HeterTLS outperformed all existing competitors on four benchmark datasets. Its effectiveness and robustness were further confirmed via ablation studies and parameter analysis.

2 Related Work

2.1 Timeline summarization

Unlike multi-document summarization (MDS), TLS executes both date selection and summary extraction (Zhou et al., 2021). In accordance with different strategies for defining the two subtasks, available approaches are categorized into three classes, whose major methods are reviewed as follows.

Direct summarization approaches (Allan et al., 2001; Yan et al., 2011a; Li and Li, 2013; Zhao et al., 2013; Suzuki and Kobayashi, 2014) treat the task as MDS with time-stamped textual summaries. Chieu and Lee (2004) directly rank and extract sentences relevant to a query from a collection of documents and place them along a timeline. As the current state-of-the-art method for direct summarization, revised submodular-function optimization, which is commonly used for MDS, is applied to search for a combination of sentences from an entire document collection (Martschat and Markert, 2018).

Date-wise summarization methods (Li et al., 2021) first select dates then extract sentences corresponding to the dates. Tran et al. (2013, 2015b) propose a supervised graphical model for selecting salient dates and tracking events on each date. In another study, text and image embeddings are jointly learned using a scalable low-rank approximation approach to generate a more readable timeline summary (Wang et al., 2016).

Event detection algorithms (Tran et al., 2015c; Pasquali et al., 2019; Duan et al., 2020) usually cluster documents by affinity propagation to detect events and summarize them individually along a timeline (Steen and Markert, 2019) or implement multi-timeline summarization (Yu et al., 2021).

2.2 Heterogeneous graph for summarization

A heterogeneous graph contains different types of nodes and multiple relationships between nodes (Xu et al., 2021; Hu et al., 2021). Wang et al. (2020) present a HAN for single or multiple document extractive summarization to enrich cross-sentence relations through additional semantic units. Jia et al. (2020) leverage a sentence-level redundancy layer into a HAN to remove excessive phrases. Although much research has gone into constructing source documents as heterogeneous graphs and using graph attention network-based first-order neighbors during information dissemination, longer-distance heterogeneous paths have not been considered. Inspired by Wang et al. (2019), we extended a HAN to TLS and developed HeterTLS to learn better node representations for downstream tasks.

3 Methodology

3.1 Problem definition and preliminaries

Given a collection of news documents $\mathcal{D}$ within $\mathcal{T}$ dates, TLS involves 1) predicting a sequence of date labels $\{y_1, \cdots, y_{\mathcal{T}} | y_i \in \{0, 1\}\}$, where $y_t = 1$ represents the t-th date included in the timeline; and 2) ranking and extracting sentences from candidates for each selected date. The number of dates as well as the length of the daily summaries are typically controlled by the user.

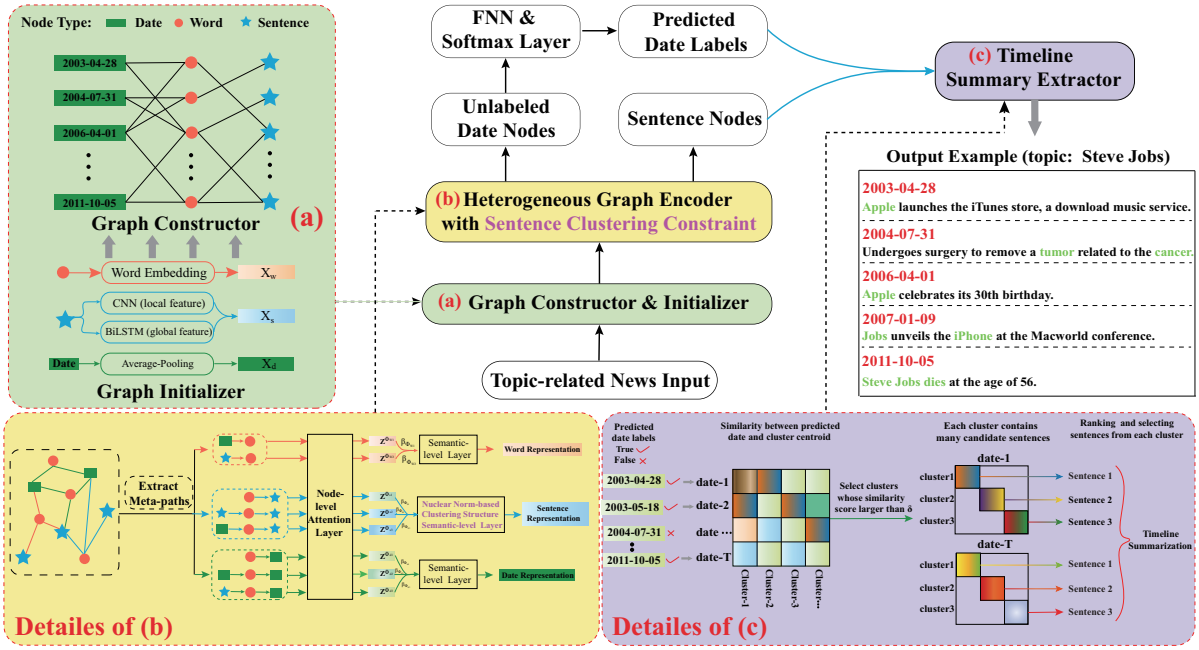

Figure 1: Model overview. HeterTLS consists of three chief components: (a) graph constructor and initializer, (b) heterogeneous graph encoder with sentence clustering constraint, and (c) timeline summary extractor. We first construct heterogeneous network for date, sentence, and word nodes with two initialization strategies. We then extract meta-paths and iteratively update node representations via HAN under nuclear norm constraint on sentence nodes. Finally, we predict unlabeled date nodes and extract sentences from candidate clusters.

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = V_d \cup V_w \cup V_s$ and $\mathcal{E} = E_{w-d} \cup E_{w-s}$, where V_d, V_w, and V_s respectively denote a node set for dates, words, and sentences and E_{w-d} and E_{w-s} are a set of undirected edges between word-date and word-sentence. Specifically, $V_d = \{d_1, \cdots, d_\mathcal{T}\}$, $V_w = \{w_1, \cdots, w_m\}$, and $V_s = \{s_1, \cdots, s_n\}$ correspond to $\mathcal{T}$ dates, m unique words, and n sentences within $\mathcal{D}$. $e_{ij} \neq 0$ $(i \in \{1, \cdots, m\}, j \in \{1, \cdots, n\})$ of E_{w-s} indicates that the i-th word appears in the j-th sentence. $e_{ij} \neq 0$ $(i \in \{1, \cdots, m\}, j \in \{1, \cdots, \mathcal{T}\})$ of E_{w-d} signifies the i-th word appears in the articles published on the j-th date. No edge exists between nodes of the same type, e.g., word pairs. We then define a meta-path and meta-path-based neighbors for disseminating information among heterogeneous nodes.

Definition 1 Meta-path Φ is defined as a path in the form of $v_1 \xrightarrow{e_1} \cdots \xrightarrow{e_q} v_{q+1}$, which describes a composite edge relation $e = e_1 \circ \cdots \circ e_q$ between nodes v_1 and v_{q+1}, where $\circ$ denotes the composition of relations.

Definition 2 Meta-path-based neighbors $\mathcal{N}_i^\Phi$ of the i-th node are defined as all nodes in a single meta-path Φ.

Figure 1 exhibits an overview of HeterTLS, which consists of three main components: (a) *graph constructor and initializer*, (b) *heterogeneous graph encoder with sentence clustering constraint*, and (c) *timeline summary extractor*. Each component is introduced subsequently in detail in the following subsections.

3.2 Graph constructor and initializer

Let $\mathbf{X}_d \in \mathbb{R}^{\mathcal{T} \times r_d}$, $\mathbf{X}_w \in \mathbb{R}^{m \times r_w}$, and $\mathbf{X}_s \in \mathbb{R}^{n \times r_s}$ respectively denote input feature matrices for date, word, and sentence nodes, where r_d, r_w, and r_s are dimensions of date representations, word embeddings, and sentence representations. We initialize the j-th sentence node in Figure 1 (a) by concatenating its local n-gram feature p_j and sentence-level global feature q_j as $X_{s_j} = [p_j; q_j]$. p_j is captured by a convolutional neural network (CNN) (LeCun et al., 1998) with different kernel sizes, and q_j is gripped by a bidirectional long short-term memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997). Considering the success of transformer-based pre-trained models, we also provide another initialization strategy: using BERT (Devlin et al., 2019) and sentence-BERT (Reimers and Gurevych, 2019) as word and sentence encoders. Date nodes take the average-pooling of their connected sentences as initialization for both aforementioned strategies.

To leverage the saliency of each word in differ-

ent sentences or dates, we propose term frequency-inverse sentence frequency (TF-ISF) and term frequency-inverse date frequency (TF-IDATEF) weights to initialize edges in E_{w-s} and E_{d-w}. Specifically, TF is the number of occurrences of w_i in s_j or d_t, and ISF/IDATEF is determined by dividing the total number of sentences or dates in $\mathcal{D}$ by the number of sentences or dates containing w_i (refer to Appendix D for more details).

3.3 Heterogeneous graph encoder with sentence clustering constraint

As Figure 1 (b) illustrates, we first iteratively update node representations via meta-paths in heterogeneous graph attention layers. We then introduce how we constrain sentence representations to reserve a low-rank-based clustering structure, which helps sentence nodes learn better event-related information. Finally, the semi-supervised date classification and sentence clustering structure are jointly learned in an overall objective.

3.3.1 Heterogeneous graph attention layer

Node representations are updated by hierarchical heterogeneous graph attention layers, where the node-level attention layer ensures information propagation and aggregation in a single meta-path, while the semantic-level one is committed to merging messages from multiple meta-paths. Specifically, referring to $\mathbf{h}_i$ as the hidden state of the i-th node, the node-level attention layer is calculated as

$$e_{ij}^{\Phi_p} = LeakyReLU(\mathbf{W}_a[\mathbf{W}_{\phi_i}\mathbf{h}_i; \mathbf{W}_{\phi_j}\mathbf{h}_j]), \quad (1)$$

$$\alpha_{ij}^{\Phi_p} = \frac{exp(e_{ij}^{\Phi_p})}{\sum_{l\in\mathcal{N}_i^{\Phi_p}} exp(e_{il}^{\Phi_p})}, \quad (2)$$

$$\mathbf{z}_i^{\Phi_p} = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j\in\mathcal{N}_i^{\Phi_p}} \alpha_{ij}^{\Phi_p}\mathbf{W}_{\phi_j}\mathbf{h}_j\Big), \quad (3)$$

where $\mathbf{W}_a$, $\mathbf{W}_{\phi_i}$, and $\mathbf{W}_{\phi_j}$ are trainable parameters, $\mathbf{z}_i^{\Phi}$ is the representation of the i-th node learned from the node-level attention layer by Φ, α_{ij}^{Φ} measures the importance of the j-th node to the i-th node via Φ, $\mathcal{N}_i^{\Phi}$ contains all nodes in single meta-path Φ, and K is the number of multi-heads.

Afterwards, the semantic-level attention layer fuses all the meta-path information for the i-th node. We extract meta-paths $\hat{\Phi}_{d_{1\sim3}}$={date-word, date-word-date, date-word-sent} for date nodes, $\hat{\Phi}_{w_{1\sim2}}$={word-sent, word-date} for word nodes, and $\hat{\Phi}_{s_{1\sim3}}$={sent-word, sent-word-sent, sent-word-date} for sentence nodes (Figure 1 (b)), while long-distance meta-paths are discarded due to their limited impact. With the assumption that the i-th node has P meta-paths as $\{\Phi_1, \cdots, \Phi_P\}$, the representation of the i-th node is updated as

$$w_{\Phi_p} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{q}^T tanh(\mathbf{W}\mathbf{z}_i^{\Phi_p} + \mathbf{b}), \quad (4)$$

$$\beta_{\Phi_p} = \frac{exp(w_{\Phi_p})}{\sum_{l=1}^{P} exp(w_{\Phi_l})}, \quad (5)$$

$$\mathbf{z}_i = \sum_{p=1}^{P} \beta_{\Phi_p}\mathbf{z}_i^{\Phi_p}, \quad (6)$$

where $\mathbf{q}$, $\mathbf{W}$, and $\mathbf{b}$ are learnable parameters and β_{Φ_p} represents the importance of the p-th meta-path for the final embedding of the i-th node.

In the same manner described by Wang et al. (2020), to avoid gradient vanishing after certain iterations, a residual connection and position-wise feed-forward network (FFN) layer with two linear transformations (Vaswani et al., 2017) are added after the semantic-level attention layer.

Iterative update: We alternately update each type of node to realize information propagation and aggregation. The updating process for the t-th iteration is measured as

$$Z_{w_{1\sim2}}^{t+1} = NLevel(H_d^t, H_s^t, H_w^t), \quad (7)$$

$$H_w^{t+1} = FFN(SLevel(Z_{w_{1\sim2}}^{t+1}) + H_w^t), \quad (8)$$

$$Z_{d_{1\sim3}}^{t+1} = NLevel(H_d^t, H_s^t, H_w^{t+1}), \quad (9)$$

$$H_d^{t+1} = FFN(SLevel(Z_{d_{1\sim3}}^{t+1}) + H_d^t), \quad (10)$$

$$Z_{s_{1\sim3}}^{t+1} = NLevel(H_d^{t+1}, H_s^t, H_w^{t+1}), \quad (11)$$

$$H_s^{t+1} = FFN(SLevel(Z_{s_{1\sim3}}^{t+1}) + H_s^t), \quad (12)$$

where $NLevel$ and $SLevel$ respectively indicate node-level and semantic-level attention layers, and H^t is the stacked hidden state of a certain type of node at the t-th timestep. Eqs. 8, 10, and 12 represent the residual connection and FFN layer.

3.3.2 Sentence clustering constraint

Detecting main events from $\mathcal{D}$ can effectively reduce the redundancy when generating summaries. Current TLS methods (Yu et al., 2021) identify major events by applying K-means directly to sentence representation matrix H_s, which has two limitations. First, K-means is sensitive to initialization and outliers, resulting in unstable outputs (Ding

and Li, 2007). Furthermore, the clustering performance is undesirable due to the independence of sentence representation learning and sentence clustering. Even though structure learning has the potential to address the above issues by co-clustering on a newly created bipartite graph to extract the clustering structure (You et al., 2021a; Nie et al., 2017), it is not suitable for our framework to make H_s a block-diagonal matrix with k components. Theorem 1 paves the way to detect the clustering structure of H_s by adding a low-rank constraint.

Theorem 1 (Chung and Graham, 1997) The multiplicity of eigenvalue 0 of the normalized Laplacian matrix of H_s is equal to the number of clusters in H_s.

Theorem 1 indicates that the block-diagonal clustering structure relies on newly constructing an adjacency network of sentence nodes, which increases the complexity of the model. Haeffele and Vidal (2020); Piao et al. (2019) propose the nuclear norm and prove that the constraint on the Laplace matrix of H_s is mathematically equal to the constraint on sentence representation matrix H_s as

$$\mathcal{L}_{cluster} = \|H_s\|_*, \tag{13}$$

where $\|H_s\|_*$ is defined as the sum of the k smallest eigenvalues, i.e., $\sum_{i=n-k}^{n} \lambda_i$ with λ_i as the i-th smallest eigenvalue (Piao et al., 2019). When $\|H_s\|_*$ is set to 0, we obtain k clusters in H_s by reorganizing its columns or rows and converting it into a block-diagonal form with k blocks, as shown in Figure 1 (c). We also determine parameter k by the elbow method (Bholowalia and Kumar, 2014).

3.3.3 Joint learning framework

Past work considered date selection and sentence clustering-based event detection as independent tasks. In HeterTLS, they are jointly trained to combine their advantages into an overall objective:

$$\mathcal{L} = \mathcal{L}_{classify} + \lambda \mathcal{L}_{cluster}, \tag{14}$$

where $\mathcal{L}_{classify}$ minimizes the cross-entropy over all labeled date nodes between the ground-truth during training, and λ serves as a weighted coefficient to balance $\mathcal{L}_{classify}$ with $\mathcal{L}_{cluster}$. Eq. 14 can be optimized via stochastic gradient descent (SGD) (Zinkevich et al., 2010) in an end-to-end manner. Readers can also refer to (Piao et al., 2019; Liu and Vandenberghe, 2009) for the detailed nuclear norm optimization strategy of $\mathcal{L}_{cluster}$.

Our date classification is trained in a transductive learning-based semi-supervised manner. We iterate all node representations in the heterogeneous graph simultaneously with 50% labeled date nodes (40% for training and 10% for verification) and 50% unlabeled date nodes as the test set. The joint learning model is able to effectively find event-based candidate clusters (see Figure 1 (c)), thereby save much running time and improve the accuracy of TLS.

	T17	Crisis	Ent.	Covid.
Topics	9	4	47	1
Timelines	19	22	47	1
Avg.Documents	508	2,310	959	26,376
Avg.Sentences	20,409	82,761	31,545	791,280
Avg.Dates	124	307	600	218
Avg.Duration	212	343	4,437	266

Table 1: Basic dataset statistics. Avg.X demonstrates average X for each topic, and Timelines refers to number of ground-truths in each dataset.

3.4 Timeline summary extractor

With l selected dates and their corresponding representations $\{\mathbf{h}_{d_1}, \cdots, \mathbf{h}_{d_l}\}$, we represent the k clusters as $\{\mathbf{h}_{c_1}, \cdots, \mathbf{h}_{c_k}\}$ by averaging sentence representations inside that cluster. As shown in Figure 1 (c), candidate clusters for the t-th selected date are determined by calculating the cosine similarity between the date representation with all cluster representations as $cos(\mathbf{h}_{d_t}, \mathbf{h}_{c_j})(j \in \{1, \cdots, k\})$. If the cosine similarity is larger than the pre-defined threshold δ (see Section 5.4), the corresponding cluster is considered a candidate for the date. Finally, we apply CENTROID-OPT (Ghalandari, 2017) as a sentence ranking algorithm within a cluster and summarize each date individually by selecting one sentence per cluster with the highest ranking score.

4 Experiments

4.1 Datasets

We carried out our experiments on the four most widely used benchmark datasets, i.e., 17 Timelines (T17) (Tran et al., 2013), Crisis (Tran et al., 2015a), Entities (Ghalandari and Ifrim, 2020), and CovidTLS (Quatra et al., 2021). All contain human-written timelines concerning certain topics, the source news articles of which are retrieved from the web at a given point in time.

Using these datasets makes it possible to comprehensively verify the effectiveness and generalization of HeterTLS because both the number of

Datasets	T17				Crisis				Entities			
Metrics	CR1-F	CR2-F	AR1-F	AR2-F	CR1-F	CR2-F	AR1-F	AR2-F	CR1-F	CR2-F	AR1-F	AR2-F
Full Oracle	0.500	0.180	0.312	0.128	0.490	0.160	0.360	0.150	0.348	0.079	0.232	0.075
CHIEU (2004)	0.290	0.072	0.067	0.019	0.374	0.070	0.052	0.012	0.275	0.053	0.036	0.011
TRAN (2013)	0.336	0.065	0.094	0.022	0.271	0.034	0.054	0.012	0.275	0.052	0.042	0.012
MARTSCHAT (2018)	0.383	0.092	0.105	0.030	0.333	0.072	0.075	0.016	0.275	0.052	0.042	0.011
DATEWISE (2020)	0.385	0.097	0.121	0.035	0.347	0.075	0.089	0.026	0.271	0.051	0.057	0.017
DASG (2021)	0.333	0.064	0.118	0.029	0.323	0.068	0.077	0.018	0.282	0.052	0.045	0.010
SDF (2021)	0.401	0.101	0.106	0.033	0.360	0.073	0.064	0.014	0.275	0.052	0.041	0.011
HeterTLS-HAN	0.398	0.101	0.141	0.052	0.372	0.070	0.092	0.026	0.272	0.052	0.054	0.015
HeterTLS-Joint	0.392	0.101	0.132	0.042	0.323	0.068	0.079	0.015	0.271	0.048	0.049	0.012
HeterTLS+Pre-trained	0.401	0.103	0.142	0.053	**0.379**†	**0.078**†	**0.107**†	**0.028**†	0.282	0.054	0.057	0.019
HeterTLS	**0.408**†	**0.108**†	**0.145**†	**0.058**†	0.374	0.075	0.105	0.028	**0.288**†	**0.058**†	**0.059**†	**0.019**†

Table 2: Concatenation- and alignment-based ROUGE-1/2 F1-scores for T17, Crisis, and Entities datasets. Best results among model-generated timelines are marked in bold. Symbol † indicates that our results significantly surpass all baselines using bootstrap test (Dror et al., 2018) with $p < 0.005$.

topics and their time spans are completely different. Specifically, the Entities dataset contains dozens of topics and spans decades per topic, while the others involve only a few topics within two years. The basic statistics are summarized in Table 1.

4.2 Evaluation metrics

In our experiments, the evaluation of model-generated timelines depended on the ROUGE metric and its variants as follows (Yu et al., 2021):

Concatenation-based ROUGE F1 Similar to conventional ROUGE, it compares a concatenated system summary with its corresponding ground-truth by referring only to the textual overlap while ignoring all time stamps of the timeline (Yan et al., 2011b; Nguyen et al., 2014; Wang et al., 2016).

Alignment-based ROUGE F1 On the basis of the above concatenation metric, it linearly penalizes the ROUGE score by the distance of date alignments (Martschat and Markert, 2017).

Date selection F1 It only measures how well the model selects dates contained in the ground-truth (Martschat and Markert, 2018).

4.3 Experimental settings

Since each topic has at least one ground-truth timeline, we considered each timeline independently if multiple ground-truths exist, and the final evaluation results were obtained by averaging scores over all timelines. We split training/verification/test sets in accordance with the ratio of 40%/10%/50% mentioned to Sec. 5.5. All experiments for a dataset were subject to leave-one-out cross-validation, and significant differences were determined by bootstrap test (Dror et al., 2018) with p-value of 0.005.

For our heterogeneous network, the vocabulary size was limited to $50,000$ and tokens were initialized with 400-dimensional GloVe embeddings (Pennington et al., 2014). We truncated an input document to a maximum length of 40 sentences and removed 10% of vocabulary with the lowest TF-IDATEF values to eliminate noise. Date/sentence nodes and edge features individually included $r_d = r_s = 128$ and 40-dimensional vectors for initialization. We set the learning rate and regularization hyper parameter λ to $5e-4$ and 1.5, respectively. Each HAN layer had 8 heads and 64-dimensional hidden size. The inner hidden size of the FFN layer was set to 512. An early stop was carried out when the validation loss did not descend for three continuous epochs. We trained all baselines as well as HeterTLS on a single Titan RTX GPU.

4.4 Baselines

The following excellent baselines were used for comparison and to demonstrate the effectiveness of HeterTLS: *direct summarization* including CHIEU (Chieu and Lee, 2004) and MARTSCHAT (Martschat and Markert, 2018); *date-wise summarization* such as TRAN (Tran et al., 2013), DATEWISE (Ghalandari and Ifrim, 2020), and SDF (Quatra et al., 2021); and *event detection* method DASG (Liu et al., 2021). We additionally follow Ghalandari and Ifrim (2020) to obtain full oracle.

5 Results and Discussion

5.1 Performance of HeterTLS

According to Tables 2 and 3, HeterTLS outperformed all baselines in terms of all metrics. Con-

Dataset Metric	T17 Date-F1	Crisis Date-F1	Entities Date-F1
Full Oracle	0.926	0.974	0.757
CHIEU (2004)	0.252	0.142	0.102
TRAN (2013)	0.517	0.289	0.185
MARTSCHAT (2018)	0.544	0.281	0.167
DATEWISE (2020)	0.544	0.295	0.205
SDF (2021)	0.553	0.302	0.397
HeterTLS-HAN	0.668	0.455	0.432
HeterTLS-Joint	0.620	0.418	0.395
HeterTLS+Pre-trained	0.688	**0.494**†	0.478
HeterTLS	**0.703**†	0.492	**0.488**†

Table 3: Date F1-scores on T17, Crisis, and Entities datasets. Bold-faced characters and † indicate best results and significant improvements over all baselines.

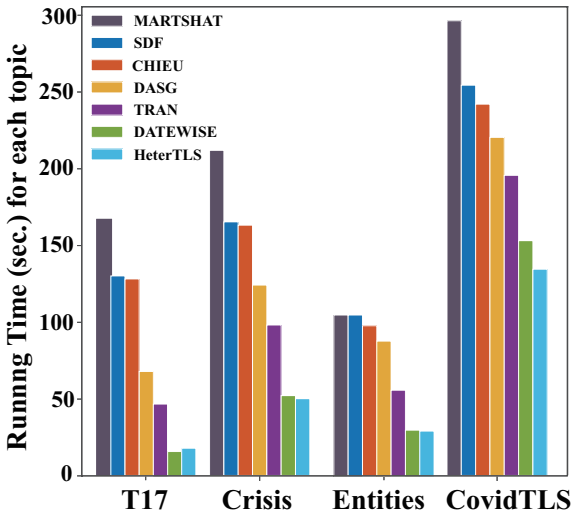

Figure 2: Comparison of running time of current state-of-the-art models and HeterTLS

sidering that DASG ignores date information, we excluded it from the Date F1 experiment. We noticed that HeterTLS with pre-trained initial node representations surpassed HeterTLS only on Crisis and CovidTLS (refer to Appendix A) datasets. This indicates that pre-trained models require larger downstream datasets (Crisis or CovidTLS datasets) to escape from the local optimum, while CNN- and Bi-LSTM-based initialization can better capture the characteristics of small-scale datasets and reach the globally optimal solution in a few epochs.

We consider three possible reasons for the excellent performance of HeterTLS. First, the HAN is configured to learn multi-level semantic features for date representations. Compared with hand-designed statistical low-level features, these features are much more distinguishable, so they improve the accuracy of date selection. Second, regarding the improvement of ROUGE scores, the introduction of low-rank-based regularization helps sentence representations learn a diagonal clustering structure, which enables HeterTLS to effectively capture the topic-related events and informative sentences. Third, date selection and sentence clustering-based event detection are jointly learned and optimized to obtain a globally optimal solution.

5.2 Ablation study

We investigated the contribution of each module to HeterTLS via ablation studies using each dataset.

HeterTLS-HAN To verify the interaction within heterogeneous connections, we show the ablation performance in Table 2 by removing the HAN and simply using unlearnable semantic features with nuclear norm constraint. We suspect that the HAN layer plays a critical role in facilitating date selection with semantic messages and providing sentence nodes with temporal clustering information, which cannot be replaced with fixed features. Meta-paths also provide abundant iterative patterns to pass semantic and temporal information.

However, sentence nodes initialized by CNN and Bi-LSTM layers help capture local and global sentence relationships, which has been proved predominant with regard to the extractive summarization task (Wang et al., 2020). Furthermore, the nuclear norm constraint can effectively reduce the redundancy between selected summary sentences. The above two components ensure the promising performance of the ablation model.

HeterTLS-joint learning Based on the assumption that the remarkable improvement of HeterTLS compared with baselines is due to jointly training node representations and clustering regularization, we show the performance in a separate learning pattern. Date representations are first learned using a HAN to predict which date should be selected to form a timeline. We then cluster sentence nodes in the graph to produce center cluster representations.

From the last block in Tables 2 and 3, implementing the subtasks individually degrades the performance to a great extent. We consider that in the joint learning framework of HeterTLS, vertices learn more discriminative features under the guidance and constraint of sentence clustering and in turn improve clustering accuracy, which cannot be imitated by separate learning. This result further indicates the superiority of HeterTLS, implying that

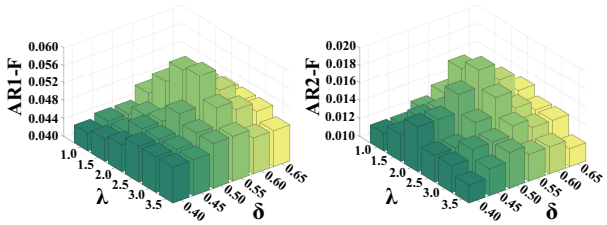

Figure 3: Impact of parameters on Entities dataset

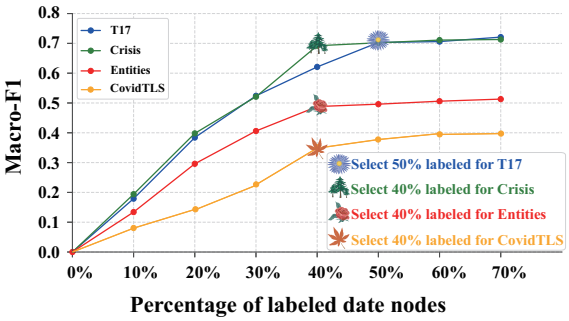

Figure 4: Ratio of labeled date nodes on training set vs. corresponding accuracy on test set

the combination of node representations and clustering structure is promising for identifying salient dates and sentence candidates simultaneously.

5.3 Running time

We conducted an investigation of running time with all models being trained with the same device and show the results in Figure 2. HeterTLS ran up to an order of magnitude faster than most baselines, while it achieved comparable running efficiency to the current fastest baseline DATEWISE (2020).

The following two reasons may explain the efficiency of HeterTLS. 1) Accurate node initialization enables the model to converge to a globally optimal solution in less than eight epochs. Since transductive semi-supervised learning requires fewer labeled date nodes, it can simplify the scale of the training model and reduce the training time caused by parameter updates. 2) Previous methods rank all candidates by measuring informativeness, redundancy, coherence, and diversity (Yan et al., 2011b). In contrast, our strategy reduces the time complexity by measuring the similarity between date and cluster representations to select candidate clusters that exceed a pre-defined threshold. It can thus extract the most informative sentence in each candidate cluster as a summary without consuming time on the multi-index optimization problem.

5.4 Impact of parameters

There are two essential hyper parameters in our experiments: λ is adopted to balance the importance between $\mathcal{L}_{classify}$ and $\mathcal{L}_{cluster}$ in Eq. 14, and δ acts as a threshold to decide the most related clusters for selected dates (Figure 1(c)). Several sets of λ and δ were tested in terms of AR1-F and AR2-F. We can clearly see the best performance from Figure 3 when $\lambda \in [2.0, 2.5]$ and $\delta = 0.55$ on Entities dataset. It is explicit that a larger δ works better. A plausible reason is that a relatively high threshold can effectively filter irrelevant clusters and reduce the redundancy of generated timelines. However, the gradual changes in histograms indicate that our method rarely fails to converge as parameters vary because it is robust and insensitive to parameters. The parameter impact on other datasets is discussed in Appendix B due to limited space.

5.5 Ratio of labeled dates

Figure 4 shows that our model achieved promising Macro-F1 scores for date classification on test sets when the ratio of labeled dates was set to 40 or 50% in the training phase. Therefore, we reasonably believe that our HAN-based transductive learning earns high-quality date classification even with small-scale labeled data, so it can be effectively applied to real TLS tasks. Specifically, HeterTLS learns high-order semantic features implied in a small amount of labeled dates, which can help predict critical time stamps that should be preserved.

5.6 Consecutive dates and redundancy

The proportions of consecutive dates in chronologically ordered model-generated timelines and ground-truth timelines were experimentally measured according to Ghalandari and Ifrim (2020). News articles and sentences published on adjacent dates tend to refer to the same story, especially in a long-time-span dataset such as Entities.

	T17	Crisis	Entities	CovidTLS
Ground-truth	0.45	0.18	0.03	0.48
MARTSCHAT	0.63	-	0.18	0.68
DATEWISE	0.62	0.52	0.30	0.66
HeterTLS	0.48	0.23	0.10	0.56

Table 4: Proportions of consecutive dates of timelines produced with different methods and ground-truths

Combining with Table 1, Table 4 reveals that because the time duration of Entities dataset is the longest, up to 12 years, the proportion of adjacent dates is the lowest among all datasets. Therefore, we reasonably believe that the trend of adjacent

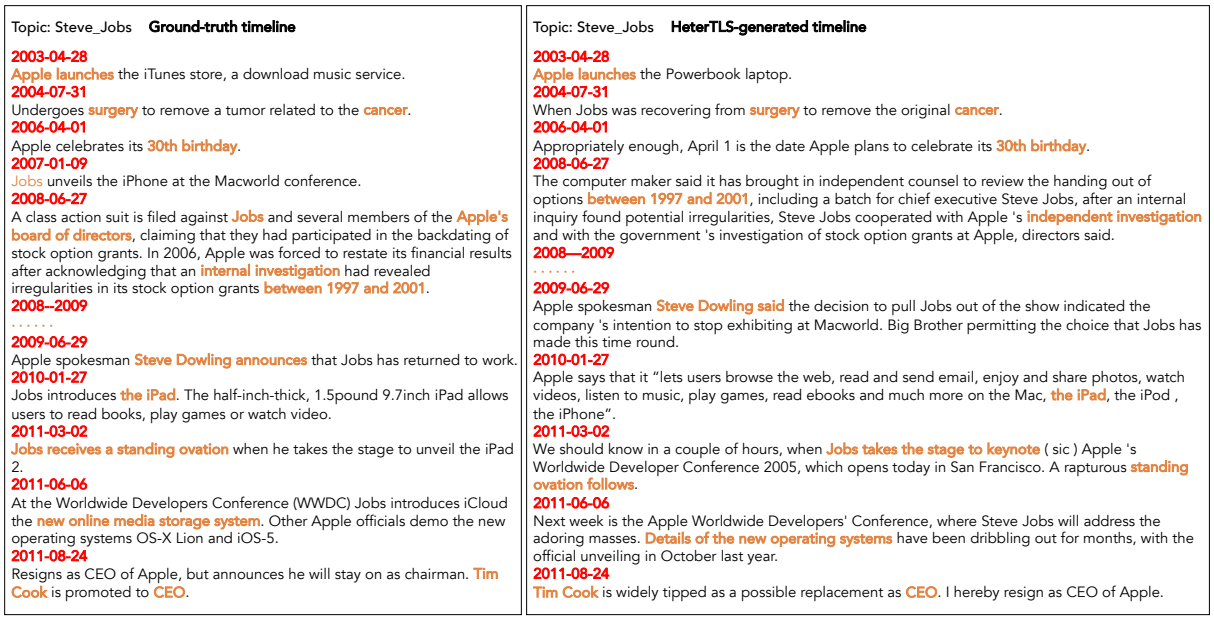

Figure 5: Partial timelines on topic of *Steve Jobs* from Entities dataset produced with ground-truth and HeterTLS

date proportion is the same as that of redundancy. The results in Table 4 indicate that HeterTLS is the closest to the ground-truth, thereby proving its ability to predict salient dates.

5.7 Case study

We now show the quality of timelines generated by HeterTLS through a cases study. The topic *Steve Jobs* is taken from Entities dataset with the time duration from 2003-04-28 to 2011-08-24. In Figure 5, parts of the ground-truth timeline of certain dates are shown on the left, while the right side lists the HeterTLS-generated timeline with similar period coverage as the ground-truth. We manually colored some keywords to illustrate consistent contents in both timeline summaries. The examples demonstrate different levels of detail in describing particular events. Three advantages of HeterTLS are explicit by comparing it with the ground-truth:

- The semi-supervised date prediction component of HeterTLS can accurately position salient dates as the ground-truth, which is the very principle for extracting TLS sentences.
- Our model can capture the major object of each event or topic well (marked in orange) in a daily summary. For example, the subject of the ground-truth on 2003-04-28 is *Apple launches*, and HeterTLS also generates the same phrase as the subject. On 2009-06-29, *Steve Dowling announces* and *Steve Dowling said* serve as subjects in the ground-truth and model-generated summary, respectively.
- Although HeterTLS generates timelines in an extractive manner, the generated summaries are short and accurate. Current extractive methods always adopt greedy or beam search to extract an uncertain number of sentences as timelines, which greatly increases redundancy. We use clustering-based constraints and intra-class extraction to ensure that HeterTLS generates short but accurate sentences.

6 Conclusion

We addressed several fundamental problems concerning TLS and proposed a joint learning model called HeterTLS, which trains a HAN by utilizing clustering structure learning-based event detection. The proposed model facilitates node representations with information of different semantic units. Meanwhile, the sentence representations with clustering structure are rich in date- and semantic-level features, which significantly reduce redundancy and improve clustering accuracy. Experimental results, including those of the ablation studies of each part of the overall architecture, demonstrated the effectiveness of HeterTLS.

Acknowledgements

We would like to gratefully thank the anonymous reviewers for their helpful comments and feedback. Jingyi You and Dongyuan Li acknowledge the support from China Scholarship Council (CSC).

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of news topics. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 10–18. ACM.

Purnima Bholowalia and Arvind Kumar. 2014. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9):17–24.

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 *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 425–432. ACM.

Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Soc., Fresno State University.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chris H. Q. Ding and Tao Li. 2007. Adaptive dimension reduction using discriminant analysis and K-means clustering. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 521–528. ACM.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. 2020. Comparative timeline summarization via dynamic affinity-preserving random walk. In *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)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1778–1785. IOS Press.

Demian Gholipour Ghalandari. 2017. Revisiting the centroid-based method: A strong baseline for multi-document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, NFiS@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 85–90. Association for Computational Linguistics.

Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the state-of-the-art in news timeline summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1322–1334. Association for Computational Linguistics.

Benjamin D. Haeffele and René Vidal. 2020. Structured low-rank matrix factorization: Global optimality, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6):1468–1482.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Yun Hu, Yeshuang Zhu, Jinchao Zhang, Changwen Zheng, and Jie Zhou. 2021. Toward fully exploiting heterogeneous corpus: A decoupled named entity recognition model with two-stage training. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1641–1652. Association for Computational Linguistics.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3622–3631. Association for Computational Linguistics.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

Jiwei Li and Sujian Li. 2013. Evolutionary hierarchical dirichlet process for timeline summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 556–560. The Association for Computer Linguistics.

Xi Li, Qianren Mao, Hao Peng, Hongdong Zhu, Jianxin Li, and Zheng Wang. 2021. Automated timeline

length selection for flexible timeline summarization. *CoRR*, abs/2105.14201.

Jingzhou Liu, Dominic J. D. Hughes, and Yiming Yang. 2021. Unsupervised extractive text summarization with distance-augmented sentence graphs. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2313–2317. ACM.

Zhang Liu and Lieven Vandenberghe. 2009. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31(3):1235–1256.

Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for timeline summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 285–290. 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, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*, pages 230–240. Association for Computational Linguistics.

Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1208–1217. ACL.

Feiping Nie, Xiaoqian Wang, Cheng Deng, and Heng Huang. 2017. Learning A structured optimal bipartite graph for co-clustering. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4129–4138.

Arian Pasquali, Vítor Mangaravite, Ricardo Campos, Alípio Mário Jorge, and Adam Jatowt. 2019. Interactive system for automatically generating temporal narratives. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part II*, volume 11438 of *Lecture Notes in Computer Science*, pages 251–255. Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Xinglin Piao, Yongli Hu, Junbin Gao, Yanfeng Sun, and Baocai Yin. 2019. Double nuclear norm based low rank representation on grassmann manifolds for clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12075–12084. Computer Vision Foundation / IEEE.

Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. Summarize dates first: A paradigm shift in timeline summarization. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 418–427. ACM.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

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.

Satoko Suzuki and Ichiro Kobayashi. 2014. On-line summarization of time-series documents using a graph-based algorithm. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 28, Cape Panwa Hotel, Phuket, Thailand, December 12-14, 2014*, pages 470–478. The PACLIC 28 Organizing Committee and PACLIC Steering Committee / ACL / Department of Linguistics, Faculty of Arts, Chulalongkorn University.

Giang Binh Tran, Mohammad Alrifai, and Eelco Herder. 2015a. Timeline summarization from relevant headlines. In *Advances in Information Retrieval - 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*, volume 9022 of *Lecture Notes in Computer Science*, pages 245–256.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 91–92. International World Wide Web Conferences Steering Committee / ACM.

Giang Binh Tran, Eelco Herder, and Katja Markert. 2015b. Joint graphical models for date selection in timeline summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the*

Asian Federation of Natural Language Processing, ACL, pages 1598–1607. The Association for Computer Linguistics.

Tuan Tran, Claudia Niederée, Nattiya Kanhabua, Ujwal Gadiraju, and Avishek Anand. 2015c. Balancing novelty and salience: Adaptive learning to rank entities for timeline summarization of high-impact events. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1201–1210. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, ACL 2020, Online, July 5-10, 2020*, pages 6209–6219. 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 *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 58–68. The Association for Computational Linguistics.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2022–2032. ACM.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3533–3546. Association for Computational Linguistics.

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, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 433–443. ACL.

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.

Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. 2021a. Robust dynamic clustering for temporal networks. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2424–2433. ACM.

Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2021b. Abstractive document summarization with word embedding reconstruction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3September, 2021*, pages 1586–1596. INCOMA Ltd.

Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. Multi-timeline summarization (MTLS): improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 377–387. Association for Computational Linguistics.

Wayne Xin Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline generation with social attention. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 1061–1064. ACM.

Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. 2021. Entity-aware abstractive multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 351–362. Association for Computational Linguistics.

Martin Zinkevich, Markus Weimer, Alexander J. Smola, and Lihong Li. 2010. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2595–2603. Curran Associates, Inc.

Appendices

A Performance of HeterTLS on CovidTLS

Metrics	CR1-F	CR2-F	AR1-F	AR2-F	Date-F
Full Oracle	0.471	0.199	0.388	0.192	0.968
CHIEU (2004)	0.203	0.021	0.008	0.001	0.176
TRAN (2013)	0.218	0.028	0.012	0.001	0.675
MARTS. (2018)	0.249	0.036	0.028	0.001	0.685
DATEWISE (2020)	0.318	0.038	0.036	0.005	0.697
DASG (2021)	0.224	0.030	0.014	0.001	0.621
SDF (2021)	0.439	0.076	0.062	0.011	0.689
HTLS-HAN	0.402	0.062	0.052	0.009	0.656
HTLS-Joint	0.388	0.058	0.048	0.006	0.648
HTLS+Pre-trained	**0.447**†	**0.078**†	**0.068**†	**0.012**†	**0.722**†
HeterTLS	0.430	0.072	0.060	0.011	0.704

Table 5: Concatenation- and alignment-based ROUGE-1/2 F1-score for CovidTLS dataset. Best results among model-generated timelines are marked in bold. Symbol † indicates that our results significantly surpass all baselines using bootstrap test (Dror et al., 2018) with $p < 0.005$.

The newly released CovidTLS dataset [1] describes the outbreak and evolution of the Covid-19 pandemic since the beginning of 2020. Because it is undoubtedly one of the most important worldwide events and affects all aspects of people's lives and work, it has been reported by an unprecedented amount and variety of news articles. The whole corpus was crawled from well-known English journals, while it is annotated with a ground-truth timeline retrieved from a public, authoritative website.

Table 5 shows the excellent performance of HeterTLS on this new dataset. Since CovidTLS is a large-scale dataset containing 26,376 documents and 791,280 sentences per topic, the pretrained node representations can escape from the local optimal solution through massive iteration processes and converge to its globally optimal solution. We are convinced that the lightweight HeterTLS is more effective for small-scale datasets while HeterTLS initialized using pre-trained language models attains better results on large-scale datasets.

B Impact of parameters on T17, Crisis, and CovidTLS Datasets

We selected several sets of λ and δ to test the performance of HeterTLS and give a general overview in Figure 6 as in Sec. 5.4. HeterTLS performed the best when $\lambda = 1.5$ and $\delta = 0.6$ on T17 dataset, while $\lambda \in [2.0, 2.5]$ and $\delta \in [0.5, 0.55]$ on Crisis and CovidTLS datasets. Even though a larger δ coupled with a smaller λ works better, HeterTLS seldom failed to converge as the parameters changed. Therefore, we reasonably believe that our proposed model is not sensitive to parameters.

[1] https://github.com/MorenoLaQuatra/SDF-TLS

C Attach date labels for sentences

Since it is a difficult problem to correctly extract the chronological order of events from time stamped-free texts, we therefore attempt to only attach dates to the sentences extracted from news articles. We assume that the first date expression detected in a sentence s is the date of the event mentioned in s. We further craft simple rules to detect date expressions in sentences and resolve them to absolute dates using the date of the article as a reference. For example, with "today" parsed as the publication date of the article, "September" and "Sunday" indicate the last September and Sunday before the article date. In the case that no date expression is detected in the entire sentences s, $date(s)$ is taken to be the publication date of the article containing s. Although this assumption is frequently incorrect in document types such as biographies, literary writings, or historical texts, we find it is reasonable for news articles. News, by definition, reports up-to-date events.

D Edge Initialization

We give a more detailed description of the edge initialization in the main body. To leverage the saliency of each word in different sentences and dates, we propose using term frequency-inverse sentence frequency (TF-ISF) and term frequency-inverse date frequency (TF-IDATEF) weights to initialize edges in E_{w-s} and E_{d-w}.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{15}$$

$$ISF_i = \log \frac{|S|}{|\{j : w_i \in s_j\}|}, \tag{16}$$

$$I\text{DATE}F_i = \log \frac{|D|}{|\{j : w_i \in d_j\}|}, \tag{17}$$

where $n_{i,j}$ indicates the number of occurrences of word w_i in sentence s_j (for edges in E_{w-s}) or date d_j (for edges in E_{d-w}), and the denominator of Eq. 15 is the sum of the number of occurrences of all words in s_j or d_j. In Eqs. 16 and 17, $|S|$ and $|D|$ respectively denote the total number of sentences

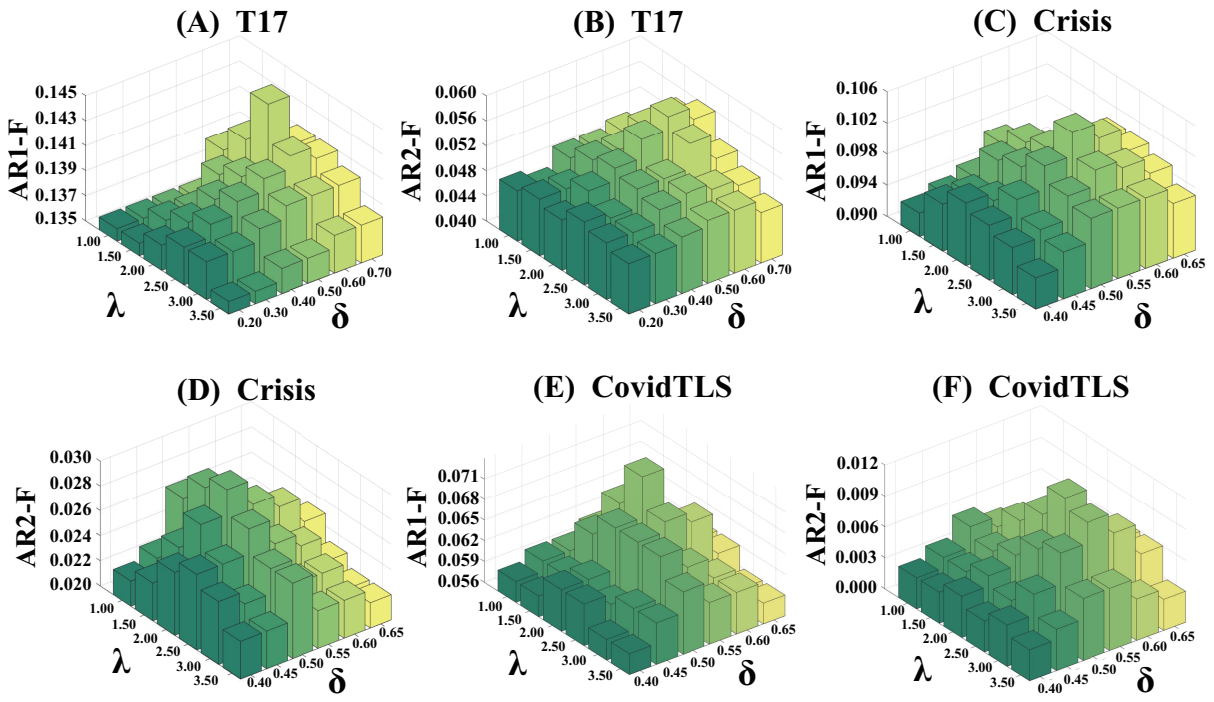

Figure 6: Impact of parameters on T17, Crisis, and CovidTLS datasets

and dates in the corpus, and $|\{j : w_i \in s_j\}|$ and $|\{j : w_i \in d_j\}|$ are the numbers of sentences or dates where term w_i appears.

Intuitively, some words, e.g., articles such as *"the"* and *"a"*, appear in many sentences and dates, while other words, e.g., *"Harry Potter"*, are not so frequent. Therefore, words with lower ISF/IDATEF values are not so important and usually have no specific meaning. Conversely, words with higher ISF/IDATEF values might be important and indicate salient information or the topic of the article. This assumption allows HeterTLS to distinguish key points from non-key points.