論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Jointly Learn Graph Node Embeddings and Graph Clustering with Temporal Information
著者(和文)	由 菁怡
Author(English)	You Jingyi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12544号, 授与年月日:2023年9月22日, 学位の種別:課程博士, 審査員:奥村 学,熊澤 逸夫,中山 実,篠﨑 隆宏,船越 孝太郎
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12544号, Conferred date:2023/9/22, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Jointly Learn Graph Node Embeddings and Graph Clustering with Temporal Information

Jingyi You

A Doctoral Thesis

Department of Information and Communications Engineering,
School of Engineering,
Tokyo Institute of Technology

Supervisor: Manabu Okumura, Professor

June, 2023

Abstract

Incorporating temporal information plays a significant role in graph clustering, summarization, and many other tasks since the inclusion of temporal information allows for a deeper understanding of the evolving patterns, trends, changes, and durations in node/edge behavior and event sequencing over time. By leveraging temporal cues, the accuracy and interpretability of clustering results can be enhanced, while a concise and coherent timeline can be constructed, presenting a more logical and comprehensive representation of the events. However, previous studies have been conservative in tackling the challenge of simultaneously learning node representations with temporal information and performing downstream clustering tasks. When node representation learning and clustering tasks are treated separately, it becomes challenging to directly incorporate specific clustering objectives into the learning process. The learned node representations may not be optimized for the specific clustering task at hand, resulting in suboptimal clustering results. Besides, temporal information plays a crucial role in understanding the evolution and dynamics of the data. Without incorporating temporal context during node representation learning, the resulting representations may not effectively capture the time-dependent patterns and relationships. To the best of our knowledge, there is currently no method that can address the aforementioned challenges in both tasks through end-to-end learning.

Our first work concentrates on dynamic community detection (or graph clustering) in temporal networks, which has attracted much attention because it is promising for revealing the underlying mechanism of complex real-world systems. Current methods are criticized for the independence of graph representation learning and graph clustering, considerable noise during temporal information smoothing, and high time complexity. We propose a **R**obust **T**emporal **S**moothing **C**lustering method (***RTSC***), which involves joint graph representation learning and graph clustering, to solve these problems. RTSC can be formulated as a constrained multi-objective optimization problem. Specifically, three-order successive snapshots are first projected into the same subspace via graph embedding. We then use the embedding matrices to learn a common low-rank block-diagonal matrix that contains

current clustering information and specific noise matrices with a sparse constraint to remove noise at each time step. To efficiently solve the challenging optimization problem, we also propose an optimization procedure based on the augmented Lagrangian multiplier (ALM) scheme. Experimental results on six artificial datasets and four real-world dynamic network datasets indicate that RTSC performs better than six state-of-the-art algorithms for dynamic clustering in temporal networks.

In the second research, we focus on addressing several challenges that plague the field of timeline summarization. Timeline summarization (TLS) is defined as a task for summarizing events in chronological order, which gives readers a comprehensive understanding of an evolutionary story. 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.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Proposals

A graph, in the context of computer science and mathematics, is a collection of objects that are interconnected. These objects are typically referred to as nodes, and the connections between them are known as edges. It is a universal language that depicts relationships, captures interactions, and visualizes complex systems. Graph-structured data are widely present in real life and in various fields, so methods utilizing graphs have significant effects in many application scenarios, such as text summarization [73], community detection [45], recommendation systems [74], and knowledge graphs [6]. To demonstrate how graphs can be applied in these applications, we take an extractive summarization method as an example. Extractive summarization is the process of identifying and selecting key phrases or sentences from the original text to form a summary, while maintaining the original context and meaning. In extractive summarization, a graph can be used by representing sentences as nodes, connecting similar sentences with edges. Then, we can apply graph-based ranking algorithms, like PageRank [53], to identify the most important sentences to include in the summary.

Recently, incorporating temporal information into a variety of static tasks has shown great success, due to the ability to reflect evolving relationships and handle time-sensitive data. This doctoral thesis primarily discusses the extension toward temporal information in traditional graph clustering and extractive text summarization tasks, namely dynamic graph clustering and timeline summarization tasks. We will now proceed to describe each task in order.

Graph clustering, which is also known as community detection [8, 30], has been used to identify tightly connected groups of vertices in networks. However, the vast majority of community detection algorithms focus only on static networks, which is insufficient to fully characterize the complex operating mechanisms in the constantly changing real world. Temporal networks are defined as a sequence of snapshots that can represent the topological evolution of entities at successive time steps. They are a powerful tool for tracking the dynamics of communities [79, 21, 75, 91, 86].

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 [9, 49, 86, 23, 87] 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. By considering temporal information, dynamic graph clustering and timeline summarization can provide more accurate and relevant results by capturing time-dependent patterns, trends, and transitions that may be missed in static approaches.

Compared with the identification of static communities, that of dynamic communities is more sophisticated because clustering accuracy and drift must be simultaneously considered. Clustering accuracy is generally used to determine whether communities can accurately reflect the topology of the current snapshot, while clustering drift quantifies the dissimilarity of communities between the current snapshot and the historical ones [88]. In terms of different strategies for balancing the clustering accuracy and drift, current community detection algorithms can be roughly categorized into the following: coupling graphs, two-stage, or evolutionary clustering methods. The first two types of methods can extend mature static community detection methods to dynamic scenes. Coupling graphs methods usually first flatten dynamic networks into a static one and then apply static community detection to identify communities [55]. Two-stage algorithms first detect communities at each time step and then match them at successive time steps [96, 64] by pre-defined mapping strategies. However, these methods always get unsatisfactory performance since they cannot make full use of the temporal information. To address this issue, evolutionary clustering methods simultaneously take into account

the clustering accuracy and drift by combining them into a weighted linear function [7, 90]. Many evolutionary clustering algorithms have been proposed: *DYNMOGA* [19], *MEGA* [20], *DECS* [42], and *MBDL* [84].

Similarly, 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 [23]: 1) *direct summarization* approaches [10, 68, 49, 17] directly identify topic-related sentences from a collection of news articles to form a timeline; 2) *date-wise summarization* methods [76, 23, 39, 60] first select salient dates and then construct a timeline for each date individually with sentences of the highest score; and 3) *event detection* algorithms [63, 17, 87] detect events by clustering sentences from multi-timeline news articles, and then identify several most important events and summarize them separately.

Although great successes have been achieved in conducting dynamic community detection and TLS, several issues remain unsolved. First, current methods treat these tasks as pipelines and adopt a two-step process, where they first extract features from original data via local or global temporal smoothing, and then use K-means or spectral clustering as a post-process for the feature matrix to obtain the final dynamic community or sentence indicator matrix. However, the manner of separately executing the two steps cannot guarantee to obtain a globally optimal solution.

In addition to this, traditional methods for each task have their own specific shortcomings. For example, during conducting clustering for dynamic networks, there is considerable noise in successive snapshots in real-world applications, resulting in the corresponding node representation matrix being corrupted. Blindly smoothing clustering information between successive snapshots without considering the noise often degrades the performance. Besides, 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 [87, 23], and employ sentence-BERT [61] 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.

To circumvent the above dilemma, we propose to jointly learn node representations and clustering structure in a graph for both dynamic graph clustering and TLS

tasks, namely **R**obust **T**emporal **S**moothing **C**lustering (RTSC) and *joint learning-based heterogeneous graph attention network for TLS* (HeterTLS). The advantage of this joint learning framework is that node representation learning can learn more discriminative features of vertices under the guidance of clustering structure and in turn improve clustering accuracy. Moreover, in RTSC, to remove the noise between successive snapshots, we learn a block-diagonal structural common embedding graph of successive snapshots via a low-rank constraint to obtain clustering information, which can more accurately facilitate the core structure of temporal networks. In HeterTLS, 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.

In our experiments, we evaluated RTSC on six artificial datasets and four real-world dynamic network datasets to investigate the effects of joint learning across different application scenes. The artificial datasets are supposed to validate the accuracy of RTSC and baseline algorithms, while real-world datasets are used to verify whether these algorithms can detect dynamic communities with particular real-world backgrounds. Besides, we also carried out our experiments for HeterTLS on four most widely used timeline benchmark datasets, i.e., 17 Timelines (T17) [68], Crisis [67], Entities [23], and CovidTLS [60]. 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. In order to demonstrate the accuracy of the extracted communities and summaries achieved by our proposals, we employed multiple evaluation metrics. Additionally, we compared our model against various baselines in terms of parameter sensitivity, ablation study, and convergence speed to showcase the overall superiority of our approach.

1.2 Contributions of This Thesis

The main contributions of this thesis are as follows.

Regarding the topic of applying joint learning of node embeddings and graph clustering to temporal networks:

- We formulate dynamic community detection as a constrained optimization problem, where node representations and graph clustering are jointly learned. Compared with other evolutionary clustering methods, RTSC is able to overcome the drawbacks of underusing temporal information and learning the above sub-tasks separately.

- To remove the noise among successive snapshots, we construct a new graph that contains the common structure of successive node representation matrices, and adopt a low-rank constraint and sparse decomposition to create the common graph having a block-diagonal clustering structure. To the best of our knowledge, RTSC can be considered as the first general common structure learning model for dynamic community detection that can more accurately facilitate the description of the core structure of temporal networks.

- The experimental results on ten temporal network datasets indicate that RTSC outperformed the state-of-the-art dynamic clustering algorithms.

Regarding the topic of constructing TLS as a heterogeneous graph to jointly perform date selection and event detection-based clustering:

- This study is the first to construct a model for automatic TLS as a heterogeneous attention network (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.

1.3 Outline of This Thesis

The rest of this thesis is organized as follows.

Chapter 2: This chapter introduces related work on graph node representation learning, community detection, timeline summarization and the application of graphs in summarization tasks.

Chapter 3: At the beginning of this chapter, we describe the dynamic graph clustering task and our joint learning method. Then, we discuss the experimental settings, results on different datasets, and analysis of our proposal.

Chapter 4: In this chapter, we first describe the existing methods for extracting timeline summaries and our proposal. Then, we introduce the models and datasets used in our experiments. Finally, we discuss our experimental results and analysis of our proposal.

Chapter 5: Finally, Chapter 5 summarizes this dissertation and discusses some directions for future work.

Chapter 2

Related Work

In the first part of this chapter, we summarize the methods that are related to graph representation learning. Then we discuss the difference with our proposal that consider clustering drift and accuracy simultaneously in the dynamic community detection task, and show previous efforts that extract and learn a common embedding matrix. In the last part, we explore prior research on the categorization of the timeline summarization task and further discuss the application of heterogeneous graph networks to the summarization task.

2.1 Node Representation Learning

Graph representation learning is used to embed vertices into a low-dimension subspace by preserving the topological structure and similarity among vertices [89], which can be used as inputs for downstream machine learning and natural language processing tasks, such as graph classification [16, 66], link prediction [95], extractive text summarization [73], and recommendation systems [74]. Many efficient and accurate algorithms have been developed for graph representation learning. For example, Deepwalk [57] preserves local topological information by sampling from a random walk and maximizing the posterior probability of the model. Based on Deepwalk, Node2vec [24] simultaneously utilizes deep-first-search and broad-first-search to learn a comprehensive representation of vertices. LINE [66] first defines the second-order similarity of vertices to present global information of graphs and then jointly learns first- and second-order similarities. O.Levy et al. [37] proved that graph embedding is equivalent to point mutual information (PMI) matrix fac-

torization, where the decomposed matrix contains low-order topological information and high-order similarity information. Current PMI matrix factorization-based graph embedding algorithms have balanced performance on all downstream tasks. Graph Convolutional Networks (GCN) [3] is a deep learning model based on graph convolution operations, which update the feature representations of nodes by aggregating the information from their neighbors. Besides, Graph Attention Networks [72] (GAT) utilize attention mechanisms and allow each node to pay varying degrees of attention to its neighboring nodes, thereby capturing the relationships between nodes more effectively.

2.2 Graph Clustering

2.2.1 Dynamic Community Detection

Community detection is a classical problem in data mining. With the increasing quantity of data in temporal networks, dynamic community detection algorithms have attracted a lot of attention [55, 96, 64]. Current dynamic community detection algorithms can be roughly categorized into three streams: coupling graphs-, two-stage-, or temporal smoothing-based methods. Coupling graphs algorithms [55] first merge vertices and edges of all snapshots into a single network and then apply static community detection [32, 26] to identify communities. Two-stage algorithms [96, 64] first independently identify static communities for each snapshot and then design rules to match the evolution of communities at successive snapshots. For example, DynaMo [96] first detects communities at each snapshot and then sets six incremental updating rules to maximize the modularity at successive snapshots.

Temporal smoothing algorithms take into account temporality during community detection. On the basis of the size of smoothing windows, these algorithms are classified as global smoothing- and local smoothing-based approaches. The former ones [43] identify the dynamic community at each time step by using all snapshots to measure the temporality of communities. An example is PisCES [43], which globally smooths the eigenvector matrix of each snapshot for temporality, achieving robust performance. Local smoothing-based methods define clustering drift in terms of how the communities reflect the previous snapshot [7, 19, 47, 42, 40, 4, 88]. Evolutionary clustering [7] presents a typical tempo-

ral smoothness framework (TSF) by balancing clustering accuracy and drift via a weighted linear function. On the basis of the core idea of TSF, many evolutionary-clustering algorithms have been proposed, where the differences lie in the definition of the clustering drift and the strategies to extract dynamic communities. For example, DYNMOGA [19] addresses dynamic community detection by reformulating the temporal smoothness as a multi-objective optimization problem and detects communities using a genetic algorithm, while sE-NMF [47] is adopted for dynamic community detection and proved to be equivalent to K-means.

2.2.2 Common Graph Learning

An integral part of RTSC is to build an accurate common embedding matrix that contained a block-diagonal clustering structure and to remove noise among successive snapshots via low-rank and sparse decomposition. However, the idea of explicitly handling the noise in multiple input matrices via the low-rank and sparse decomposition is not new. For example, the robust multi-view clustering method [80] separates the noise in multiple input matrices via learning a low-rank transition probability matrix. Nie et al. [52] proposed to learn a new probability-based similarity matrix with a low-rank constraint and to add $l_{2,1}$ norm to remove the noise of data.

RTSC shares some of the features with the previous methods for low-rank and sparse decomposition. However, we adopt a low-rank constraint to build our common graph embedding matrix having a block-diagonal clustering structure, and apply sparse decomposition to make our common embedding matrix more informative and the specific noise matrix at each time step as sparse as possible.

2.3 Text Summarization

2.3.1 Timeline summarization

Unlike multi-document summarization (MDS), TLS executes both date selection and summary extraction [94]. 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 [1, 82, 38, 92, 65] treat the task as MDS with time-stamped textual summaries. [10] 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 [49].

Date-wise summarization methods [39] first select dates then extract sentences corresponding to the dates. [68, 69] 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 [76].

Event detection algorithms [70, 54, 17] usually cluster documents by affinity propagation to detect events and summarize them individually along a timeline [63] or implement multi-timeline summarization [87].

2.3.2 Heterogeneous graph for summarization

A heterogeneous graph contains different types of nodes and multiple relationships between nodes [81, 29]. [73] present a HAN for single or multiple document extractive summarization to enrich cross-sentence relations through additional semantic units. [31] 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 [77], we extended a HAN to TLS and developed HeterTLS to learn better node representations for downstream tasks.

Chapter 3

Conducting joint learning on dynamic graph clustering

3.1 Methodology

3.1.1 Preliminaries

We consider a graph is denoted as $G = (V, E)$, where the associated vertex (or node) set and edge set are represented as $V = \{v_1, \cdots, v_n\}$ and $E = \{(v_i, v_j) | v_i, v_j \in V\}$, respectively. W is the weighted adjacent matrix of G, whose element w_{ij} denotes the weight on edge (v_i, v_j). The degree sequence matrix is denoted as $D = diag(d_i, \cdots, d_n)$, where d_i is the degree of v_i, i.e., $d_i = \sum_j w_{ij}$. We use $\|W\|$ and W' to denote the Frobenius norm and the transpose matrix of W.

Let $\{1, \cdots, \tau\}$ be the set of time steps. Temporal network $\mathcal{G} = \{G_1, \cdots, G_\tau\}$ is a sequence of snapshots evolving over time, where G_t is derived from G_{t-1} and will evolve into G_{t+1}. The adjacent matrix of $\mathcal{G}$ is $\mathcal{W} = \{W_1, \cdots, W_\tau\}$.

3.1.2 Problem Definition

Static community detection in G aims to obtain a partition of V, denoted as $\{C_i\}_{i=1}^k$, with the restriction of $V = \bigcup C_i$ and $C_i \bigcap C_j = \emptyset$ for $i \neq j$, where C_i is the i-th community and k is the total number of communities. While differently, dynamic community detection constructs a partition at each time step, denoted as $\{C_{it}\}_{i=1}^{k_t}$, where C_{it} is the i-th dynamic community at time step t. Therefore, dynamic community detection should simultaneously take into account both clustering accu-

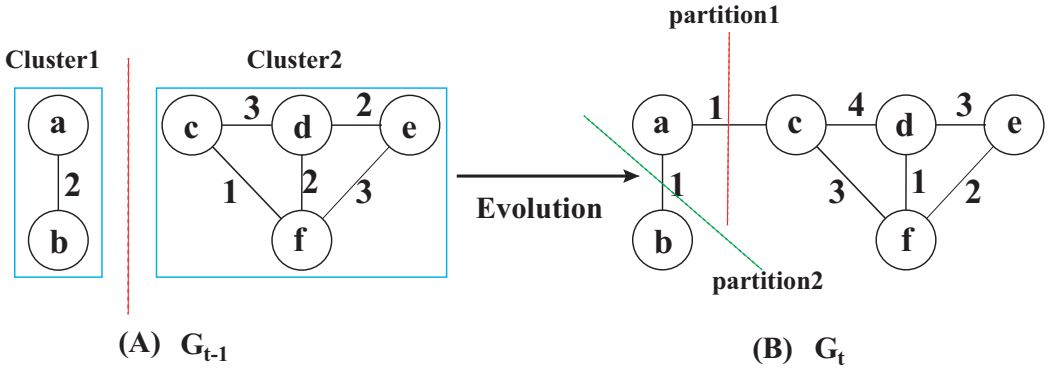

Figure 3.1: A schematic example of dynamic communities evolution, where (A) G_{t-1} contains two communities $C_{1,t-1} = \{a, b\}$ and $C_{2,t-1} = \{c, d, e, f\}$, (B) G_t has two types of clustering pattern generated by the red and green lines.

racy and drift among snapshots for different time steps. A schematic example is shown in Fig.3.1, where G_{t-1} contains two communities, $C_{1,t-1} = \{a, b\}$ and $C_{2,t-1} = \{c, d, e, f\}$. In G_t, community structures created with the red line and the green line are equal in terms of cutting cost. However, communities created with the red line are better than those created with the green line since {a,b} is connected in G_{t-1}. Thus, dynamic community detection is used to discover community structures $\{C_{it}\}_{i=1}^{k}$ at time step t, where $\{C_{it}\}_{i=1}^{k}$ simultaneously maximizes clustering accuracy and minimizes clustering drift.

3.1.3 Proposed Method

In this subsection, we introduce the framework of our **R**obust **T**emporal **S**moothing **C**lustering method (RTSC) in detail. Its optimization rules and algorithm analysis are discussed later.

An overview of RTSC is illustrated in Fig.3.2. RTSC consists of three major components: node representation learning, common embedding matrix learning, and low-rank constraint-based block-diagonal clustering for the common parts. Therefore, the overall objective function of RTSC is included in the three major components corresponding to the above as well. We use the multi-objective optimization strategy to obtain better graph clustering results while achieving promising graph embeddings simultaneously.

First, node representation learning is used to adopt low-dimension features to represent the original complex graph structure data. Qiu et al. [59] proved that

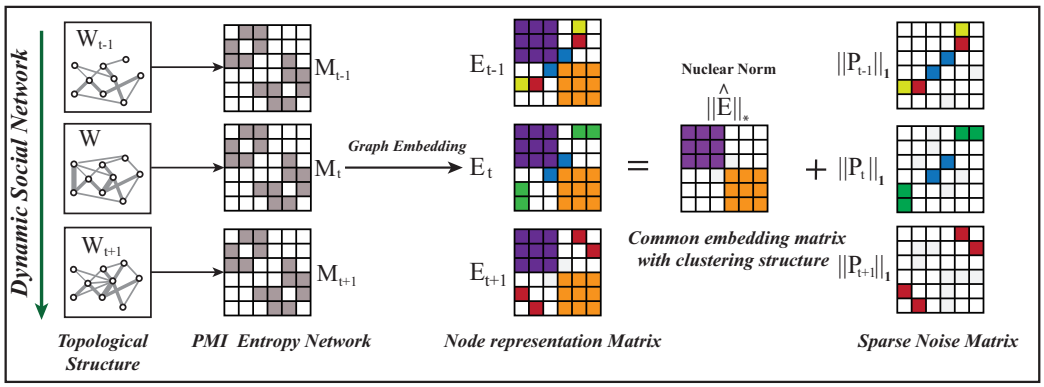

Figure 3.2: An overview of the proposed algorithm, which consists of graph embedding, common embedding matrix learning, and clustering structure learning. Given three successive snapshots, we first translate them into PMI matrices, and then decompose these snapshots and get node representation matrices. After that, we learn a common graph embedding matrix by adding the l_1 norm on noise matrices. Finally, we add a nuclear norm on the common embedding matrix to make it a block-diagonal matrix with a clustering structure.

node representation learning is equivalent to matrix factorization, implying that matrix factorization is an effective alternative for graph embedding. Therefore, it is common and convenient to directly factorize the adjacent matrix W_t of G_t through (non-negative matrix factorization) NMF [36], i.e.,

$$\mathcal{O}(G_t) = \|W_t - B_t E_t\|^2, \quad s.t.\ B_t \geq 0,\ E_t \geq 0, \tag{3.1}$$

where $B_t \in R^{n\times l}$ and $E_t \in R^{l\times n}$ are the basis matrix and feature matrix, respectively. However, the adjacent matrix solely describes the direct interactions between pairs of vertices, which cannot fully capture the underlying information of long-distance or unconnected node pairs. Levy et al. [37] proved that matrix factorization of PMI matrices is equivalent to skip-gram-based graph embedding, which can preserve both the low-order topological information and high-order unconnected long-distance information. Then, we rewrite Eq.(3.1) as:

$$\mathcal{O}(G_t) = \|M_t - B_t E_t\|^2, \quad s.t.\ B_t \geq 0,\ E_t \geq 0,\ B_t^T B_t = I, \tag{3.2}$$

where columns of $E_t \in R^{l\times n}$ are desired representations for vertices, l is the dimension of vertices, B_t contains the representation of basis vectors, I is the identity

matrix, and PMI matrix M_t of G_t is defined as:

$$m_{ij} = max\{log\frac{w_{ij}\sum_k d_k}{d_i d_j - \kappa}, 0\}, \tag{3.3}$$

where κ is a hyper-parameter controlling the number of samples via negative sampling, which is always set to 2.

Second, the selected features in successive time steps might be corrupted by noise, which will result in a small portion of data points being assigned to wrong clusters. On the basis of this assumption, each graph embedding matrix E_t can be naturally decomposed into two parts: a shared latent common embedding matrix $\hat{E}_t$, that reflects the inherent structure of successive snapshots ($\{G_{t-1}, G_t, G_{t+1}\}$ in this paper which can get best smoothing performance in our experiments), and a deviation error matrix P_t, that encodes the noise in E_t:

$$E_t = \hat{E}_t + P_t, \quad t = \{1, \cdots, \tau\}. \tag{3.4}$$

Since we assume the selected common features in successive time steps are sufficient to identify most of the clustering structure, it is reasonable to consider that there is only a small fraction of elements in E_t that significantly differ from the corresponding ones in $\hat{E}_t$. Therefore, we add l_1 norm on P_t to make it as sparse as possible:

$$\mathcal{O}(P_t) = \sum_{i=t-1}^{t+1} \|P_i\|_1 \quad s.t.\ E_i = \hat{E}_i + P_i. \tag{3.5}$$

Finally, current community detection algorithms identify the community structure at t using K-means on the basis of E_t directly, which has two limitations. On one hand, K-means is sensitive to the initialization, resulting in unstable outputs. On the other hand, performance of clustering can be undesirable because of the independence of graph representation learning and graph clustering. Even though structure learning has the potential to address above-mentioned issues by preserving specific structural information [51], it is still difficult to make $\hat{E}_t$ a block-diagonal matrix with k-connected components. Theorem 1 proves that it is equivalent to the rank of the matrix, which can be efficiently solved by adding a low-rank constraint.

Theorem 1 [11] The multiplicity of eigenvalue 0 of normalized Laplacian ma-

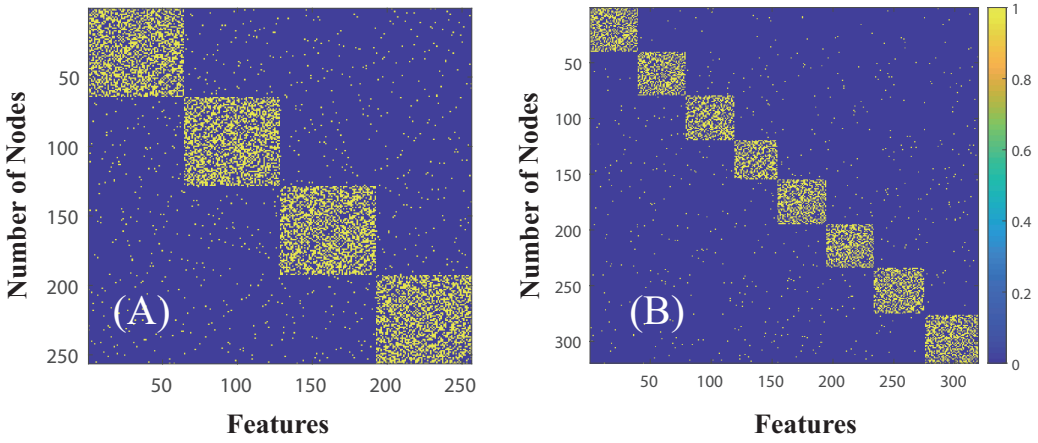

Figure 3.3: Visualized block-diagonal clustering structures of the SYN-VAR dataset (see Sec.3.3) by RTSC at (A) time step 1 and (B) time step 5.

trix $L_{E_t} = I - D_t^{-1/2} E_t D_t^{-1/2}$ is equal to the number of connected subgraphs in E_t, where D_t is the diagonal matrix with degree sequence of E_t.

Based on Theorem 1, the block-diagonal clustering structure learning can be transformed into a matrix rank problem, i.e., calculating a nuclear norm [25], which is formulated as

$$\mathcal{O}(\hat{E}_t) = \|\hat{E}_t\|_* \quad s.t.\ E_t = \hat{E}_t + P_t, \tag{3.6}$$

where $\|E_t\|_*$ is the sum of the k smallest eigenvalues, i.e., $\sum_{i=n-k}^{n} \lambda_i$. When the sum of the k smallest eigenvalues is set to 0, we can get k-connected clusters in $\hat{E}_t$ by reorganizing its columns/rows and convert it into a block-diagonal form with k blocks (as shown in Fig.3.3). The rank of this block-diagonal matrix is not greater than k.

Combining Eqs.(3.2,3.5,3.6), RTSC solves the following optimization problem:

$$min \quad \mathcal{O}(G_t) + \mathcal{O}(\hat{E}_t) + \alpha \mathcal{O}(P_t), \tag{3.7}$$

which can be formulated as:

$$\begin{aligned} &\min_{B, E_i, \hat{E}_t, P_i} \sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2 + \|\hat{E}_t\|_* + \alpha \sum_{i=t-1}^{t+1} \|P_i\|_1 \\ &s.t. \quad E_i = \hat{E}_t + P_i, \quad B^T B = I, \quad E_i \geq 0, \quad B \geq 0, \quad \hat{E}_t^T \vec{\mathbf{1}} = 1, \end{aligned} \tag{3.8}$$

where α is a hyper-parameter to balance the noise removal. In order to learn a

better common embedding matrix in one subspace, the three successive snapshots are forced to share a common basis vector matrix B. Additionally, we apply orthogonal constraints on B to minimize the feature redundancy between nodes in the embedding process.

3.2 Optimization

The optimization problem in Eq.(3.8) is hard to solve because it contains non-convex non-negative matrix factorization and $\hat{E}_t$ has a nuclear norm constraint. In this section, we introduce our proposed optimization procedure based on the ALM scheme, which can accelerate the convergence of RTSC by giving a closed-form solution for each variable.

By introducing an auxiliary variable Q, we convert Eq.(3.8) into the following equivalent form:

$$\begin{aligned} &\min_{B,E_i,Q,\hat{E}_t,P_i} \sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2 + \|Q\|_* + \alpha \sum_{i=t-1}^{t+1} \|P_i\|_1 \\ &s.t.\ E_i = \hat{E}_t + P_i,\ B^T B = I,\ E_i \geq 0,\ B \geq 0,\ \hat{E}_t^T \vec{\mathbf{1}} = 1,\ \hat{E}_t = Q. \end{aligned} \tag{3.9}$$

The corresponding augmented Lagrange function of Eq.(3.9) is:

$$\begin{aligned} \mathcal{L} = &\sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2 + \|Q\|_* + \alpha \sum_{i=t-1}^{t+1} \|P_i\|_1 \\ &+ \sum_{i=t-1}^{i+1} < Y_i, \hat{E}_t + P_i - E_i > + \frac{\mu}{2} \sum_{i=t-1}^{t+1} \|\hat{E}_t + P_i - E_i\|_F^2 \\ &+ < Z, \hat{E}_t - Q > + \frac{\mu}{2} \|\hat{E}_t - Q\|_F^2 \\ &s.t. \quad B^T B = I, \quad E_i \geq 0, \quad B \geq 0, \quad \hat{E}_t^T \vec{\mathbf{1}} = 1, \end{aligned} \tag{3.10}$$

where Z and Y_i represent the Lagrange multipliers, ¡·, ·¿ denotes the inner product of matrices, and $\mu > 0$ is an adaptive penalty parameter. Then, we will present the update rules for B, E_i, Q, $\hat{E}_t$, and P_i, by minimizing $\mathcal{L}$ in Eq.(3.10) with other variables being fixed.

3.2.1 Updating B

Leaving other variables unchanged, the sub-problem for optimizing B is

$$\min_{B} \sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2, \tag{3.11}$$

The only variable in Eq.(3.11) is B. We expand Eq.(3.11) with polynomials as follows:

$$\begin{aligned}
&\min_{B} \sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2, \\
&= \min_{B} \sum_{i=t-1}^{t+1} (M_i^T M_i - M_i^T BE_i - E_i^T B^T M_i + E_i^T B^T BE_i) \\
&= \min_{B} \sum_{i=t-1}^{t+1} (M_i^T M_i - M_i^T BE_i - E_i^T B^T M_i + E_i^T E_i)
\end{aligned} \tag{3.12}$$

By removing the constants $M_i^T M_i$ and $E_i^T E_i$ and adding $E_i M_i^T M_i E_i^T$ and I, Eq.(3.12) is transformed as:

$$\begin{aligned}
&= \min_{B} \sum_{i=t-1}^{t+1} (-M_i^T BE_i - E_i^T B^T M_i) \\
&= \min_{B} \sum_{i=t-1}^{t+1} (E_i M_i^T M_i E_i^T - M_i^T BE_i - E_i^T B^T M_i + I) \\
&= \min_{B} \sum_{i=t-1}^{t+1} (E_i M_i^T M_i E_i^T - M_i^T BE_i - E_i^T B^T M_i + B^T B) \\
&= \min_{B} \sum_{i=t-1}^{t+1} \|M_i E_i^T - B\|_F^2.
\end{aligned} \tag{3.13}$$

Therefore, the optimal solution for B is formulated as [62]

$$B = U_B V_B, \tag{3.14}$$

where U_B and V_B are left and right singular matrices of the economic singular value decomposition (SVD) of $\frac{1}{3}\sum_{i=t-1}^{t+1} M_i E_i^T$.

3.2.2 Updating E_i

By fixing other variables and removing irrelevant items for E_i, Eq.(3.10) is formulated as

$$\mathcal{L}(E_i) = \min_{E_i} \sum_{i=t-1}^{t+1} \|M_i - BE_i\|_F^2 + \frac{\mu}{2} \sum_{i=t-1}^{t+1} \|\hat{E}_t + P_i - E_i + \frac{Y_i}{\mu}\|_F^2. \quad (3.15)$$

Since Eq.(3.15) is convex in terms of E_i, the partial derivative $\frac{\partial \mathcal{L}(E_i)}{\partial E_i}$ can be deduced as

$$\frac{\partial \mathcal{L}(E_i)}{\partial E_i} = \sum_{i=t-1}^{t+1} (2E_i - 2B^T M_i - \mu(\hat{E}_t + P_i + \frac{Y_i}{\mu} - E_i)). \quad (3.16)$$

In accordance with the Karush-Kuhn-Tucker condition, we set $\frac{\partial \mathcal{L}(E_i)}{\partial E_i} = 0$ and obtain the update rule for E_i as

$$E_i = \frac{2B^T M_i + \mu(\hat{E}_t + P_i + \frac{Y_i}{\mu})}{2 + \mu}. \quad (3.17)$$

3.2.3 Updating Q

When other variables are fixed, the sub-problem w.r.t. Q is

$$\min_{Q} \|Q\|_* + \frac{\mu}{2} \|\hat{E}_t - Q + \frac{Z}{\mu}\|_F^2, \quad (3.18)$$

which can be solved using the singular value threshold method [5]:

$$Q = U S_{1/\mu}(\Sigma) V^T, \quad (3.19)$$

where $S_\delta(X)$=max($X - \delta$,0)+min($X + \delta$,0) is the shrinkage operator.

3.2.4 Updating P_i

The sub-problem w.r.t. P_i can be simplified as

$$\min_{P_i} \alpha \|P_i\|_1 + \frac{\mu}{2} \|P_i - (E_i - \hat{E}_t - \frac{Y_i}{\mu})\|_F^2, \quad (3.20)$$

which has a closed-form solution $P_i = S_{\alpha/\mu}(E_i - \hat{E}_t - \frac{Y_i}{\mu})$.

3.2.5 Updating $\hat{E}_t$

With other variables being fixed, we update $\hat{E}_t$ by solving

$$\begin{aligned} \hat{E}_t = \underset{\hat{E}_t}{argmin} \frac{\mu}{2} \sum_{i=t-1}^{t+1} \|\hat{E}_t + P_i - E_i + \frac{Y_i}{\mu}\|_F^2 + \frac{\mu}{2}\|\hat{E}_t - Q + \frac{Z}{\mu}\|_F^2 \\ s.t. \quad \hat{E}_t \geq 0, \quad \hat{E}_t^T \vec{\mathbf{1}} = 1. \end{aligned} \tag{3.21}$$

For ease of presentation, we define

$$F = \frac{1}{m+1}(Q - \frac{Z}{\mu} + \sum_{i=t-1}^{t+1}(E_i - P_i - \frac{Y_i}{\mu})), \tag{3.22}$$

and then rewrite Eq.(3.21) as

$$\begin{aligned} \hat{E}_t &= \underset{\hat{E}_t}{argmin} \frac{1}{2}\|\hat{E}_t - F\|_F^2, \ s.t. \hat{E}_t \geq 0, \hat{E}_t \vec{\mathbf{1}} = 1 \\ &= \underset{\hat{E}_{t(1)}, \cdots, \hat{E}_{t(n)}}{argmin} \frac{1}{2}\sum_{i=1}^{n} \|\hat{E}_{t(i)} - F_i\|_F^2, \ s.t. \sum_{j=1}^{l} \hat{E}_{t(ij)} = 1, \hat{E}_{t(ij)} \geq 0, \end{aligned} \tag{3.23}$$

where $\hat{E}_{t(i)}$ and F_i denote the i-th row of the matrix $\hat{E}_t$ and F, respectively. According to [18], Eq.(3.23) can be decomposed into n-independent sub-problems:

$$\underset{\hat{E}_t}{argmin} \frac{1}{2}\|\hat{E}_{t(i)} - F_i\|_2^2, \ s.t. \sum_{j}^{l} \hat{E}_{t(ij)} = 1, \hat{P}_{ij} \geq 0. \tag{3.24}$$

Each sub-problem is a proximal operator problem with a probabilistic simple constraint, which can be efficiently solved using the projection algorithm [18]. Since Eq.(3.10) is convex subject to linear constraints, and all its sub-problems have closed-form solutions, RTSC converges to the global optimum with a linear convergence rate. The flow of RTSC is shown in Algorithm 1.

Algorithm 1 RTSC

Require:

$\mathcal{G}$: Temporal networks $(G_1, \ldots, G_\tau)$;

k_t: Number of dynamic communities at time step t;

α: A parameter controls the importance of sparsity.

Ensure:

$\{C_{it}\}_{i=1}^{k_t}$: Dynamic communities at t;

Initialize matrix B, E_i, $\hat{E}_t$, P_i, Q, $\mu = 10^{-6}$, $\alpha = 1.9$, $max_\mu = 10^{10}$, $\epsilon = 10^{-8}$;
Construct the PMI matrix M_i $(i = 1, \cdots, \tau)$ for each t.
repeat
  Set $F \leftarrow \frac{1}{m+1}(Q - \frac{Z}{\mu} + \sum_{i=t-1}^{t+1}(E_i - P_i - \frac{Y_i}{\mu}))$.
  for $i = t-1$ to $t+1$ **do**
    Take F_j as input to update $\hat{E}_{t(j)},\ j \in \{1, \cdots, n\}$
  end for
  Fix the other variables, update B in accordance with Eq.(3.14);
  Fix the other variables, update E_i in accordance with Eq.(3.17);
  Fix the other variables, update Q in accordance with Eq.(3.19);
  Fix the other variables, update P_i according Eq.(3.20);
  Fix the other variables, update $\hat{E}_t$ according to Eq.(3.24);
until $\{\|\hat{E} + P_i - E_i\|_\infty, \|\hat{E} - Q\|_\infty\} \leq \epsilon$;
According to diagonal block structure, discover dynamic communities on the basis of $\hat{E}_t$ to get $\{C_{it}\}_{i=1}^{k_t}$.
return

3.2.6 Algorithm Analysis

Space Complexity Storing M_t, B and E_t requires space O(τn^2), O(nl) and O(nl), respectively. The space complexity for both common embedding matrix $\hat{E}_t$ and specific noise matrix P_i is also O(nl). Therefore, the overall space complexity of RTSC is O(τn^2).

Time Complexity For each time step t, the construction of M_t requires time O(n^2). Updating matrix B, E_i and Q has time complexity of O($n^2 l$), while P_i and $\hat{E}_t$ are O(1). Thus, the total time complexity is O($n^2 l\theta$), where θ is the number of

iterations in the algorithm.

3.3 Experiments

We conducted experiments to validate the superiority of RTSC.

3.3.1 Datasets

Table 3.1 summarizes the statistics of six artificial and four real-world temporal network datasets. The artificial datasets are supposed to validate the accuracy of RTSC and baseline algorithms, while real-world datasets are used to verify whether these algorithms can detect dynamic communities with particular real-world backgrounds.

SYN-FIX/SYN-VAR Both the two datasets originate from the static GN network [33] by incorporating dynamics. Specifically, *SYN-FIX* initially contains 128 vertices and 4 communities, where each community contains 32 vertices. Three vertices are randomly selected from each community and are assigned to other communities at each time step. This is iterated 10 times with a fixed number of communities. *SYN-VAR* is a more complicated temporal network dataset that selects eight vertices at each time step to form a new community. The above process is repeated five times, and then all vertices return to their own community in the next five times.

Table 3.1: Statistics of temporal network dataset used in this study.

	Network	$\|V\|$	$\|E\|$	τ
Artificial Temporal Networks	SYN-FIX	128	20,480	10
	SYN-VAR	256	59,526	10
	Switch	8,000	20,420,300	20
	Birth-Death	8,000	20,574,452	20
	Merge-Split	8,000	19,170,256	20
	Expand-Contract	8,000	18,834,234	20
Real-world Temporal Networks	Cellphone	400	10,248	10
	Email	151	7,273	12
	ca-cit-HepTh	22,900	2,700,000	10
	Facebook	63,700	13,000,000	10

Greene Dataset *SYN-FIX* and *SYN-VAR* are insufficient to fully validate the performance of the algorithms because the patterns of dynamic communities are relatively simple and the network size is small. Therefore, we employed the Greene dataset [78] to better characterize different change scenarios, which contains four evolution events of *Switch*, *Birth-Death*, *Merge-Split*, and *Expand-Contract*. 10% of vertices in the *Switch* dataset exchange their communities at each time step. In the *Birth-Death* dataset, 10% of new communities are created by randomly selecting vertices from existing communities, and 10% of existing communities are removed at each time step. In the *Merge-Split* dataset, 10% of the communities are split to two communities and 10% of the communities are merged into one community at each time step, while 10% of communities in the *Expand-Contract* dataset expand or contract 75% of their original size at each time step.

Real-world Temporal Network Datasets The *Cellphone* dataset, developed from the VAST 2008 mini challenge[1], consists of cellphone call records among the members of the fictitious Paraiso movement. In this dataset, phone calls between 400 cellphones were recorded for ten days in June 2006. Nodes represent people and an edge represents the occurrence of a phone call between two people. These records are divided into ten equal sizes to build temporal networks. The *Enron Email* (hereafter, *Email*) dataset contains the email communications[2] of the U.S. enterprise Enron in 2001. It is split into 12 snapshots by month, each snapshot consisting of 151 nodes (or users) and edges representing messages sent from one user to another. The *ca-cit-HepTh* dataset[3] is collected from arXiv and covers all the citations in it with the regulation below. Edges from u to v indicate that a paper u cites another paper v, but the dataset does not contain any information on it if a paper cites or is cited by a paper outside the dataset. The *Facebook friendship graph* (hereafter, *Facebook*)[4] dataset contains 63,700 nodes and more than ten million edges to represent users and the relationship between two users, respectively. This dataset is divided into ten snapshots by month.

[1] http://www.cs.umd.edu/hcil/VASTchallenge08/
[2] http://www.cs.cmu.edu/ enron/
[3] networkrepository.com/dynamic.php
[4] networkrepository.com/fb-wosn-friends.php

3.3.2 Metrics

Three metrics, normalized mutual information (NMI) [12], accuracy, and purity [93], were used to measure the performance of both RTSC and baselines. Given C and C^* as the predicted community structure and ground truth respectively, a confusion matrix N is constructed, where its element n_{ij} represents the number of vertices overlapped by the i-th community in C^* and j-th community in C. NMI is defined as

$$NMI(C, C^*) = \frac{-2\sum_{i=1}^{|C|}\sum_{j=1}^{|C^*|} N_{ij} log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_{i=1}^{|C|} N_{i.} log(\frac{N_{i.}}{N}) + \sum_{j=1}^{|C^*|} N_{.j} log(\frac{N_{.j}}{N})}. \tag{3.25}$$

Accuracy is used to measure the percentage of obtained correct labels with the definition of

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} \delta(z_i, g_i), \tag{3.26}$$

where n is the total number of vertices, z_i and g_i denote the predicted and ground truth label of the i-th vertex, and $\delta(x, y)$ is an indicator function that equals 1 when $x = y$, and equals 0 otherwise.

Purity [93] measures the extent to which each cluster contains vertices primarily from one class and is defined as

$$Purity = \sum_{i=1}^{k} \frac{n_i}{n} P(C_i), \quad P(C_i) = \frac{1}{n_i} max_j(n_i^j), \tag{3.27}$$

where C_i is a cluster with size n_i, n_i^j is the number of vertices of the i-th input class that are classified into the j-th cluster, k denotes the number of clusters determined by Elbow method [41] for each time step, and n is the total number of vertices.

3.3.3 Baselines

To fully validate the performance of RTSC, six excellent models, PisCES [43], DYNMOGA [19], sE-NMF [47], dynamic network embedding (DNE) [16], DECS [42], and MEGA [20], were employed as baselines, covering global smoothing, matrix factorization-based local smoothing, and deep learning-based dynamic community detection. We selected and applied PisCES for comparison since it is a typical global smoothing dynamic community detection algorithm, and DYNMOGA is

the state-of-the-art multi-objective optimization-based model. sE-NMF and DECS are two well-known local smoothing dynamic community detection methods, while DNE and MEGA are state-of-the-art deep learning-based dynamic network embedding algorithms. We fine-tuned all the parameters in these baselines for fairness. We also ran them twenty times on each dataset and took the averages as the final experimental results.

3.4 Results and Discussion

3.4.1 Accuracy

According to [19], we apply the DYNMOGA algorithm several times to detect the community structure with the highest soft modularity score as the ground truth label on real-world datasets. The average NMI, accuracy, and purity of the algorithms on all temporal network datasets are shown in Table 3.3, 3.2, and 3.4, where RTSC outperformed the others in terms of average NMI, followed by DECS and DYNMOGA. Specifically, the improvement on NMI with RTSC ranges from 0.4% on *Birth-Death* to 9.3% on *Cellphone*. However, NMI can hardly measure the internal structure of the community since it will increase as the size of the community expands. Thus, we used accuracy and purity to measure the internal structure quality of clustering results, as shown in Table 3.3, 3.2, and 3.4. These results indicate that RTSC is also superior to the others in terms of accuracy and purity, where the improvement in accuracy ranged from 0.7% on *SYN-VAR* to 27.3% on *Email*, and that in purity ranged from 1.2% on *Birth-Death* to 36.4% on *Cellphone*. However, we found that DYNMOGA outperforms others on *Birth-Death*. We consider that it is because DYNMOGA can translate dynamic community detection into a multi-objective optimization problem to catch the global optimal solution for clustering. DECS achieved the highest accuracy on *Email* because it develops a migration operator to ensure that nodes are grouped together and adopts a genome matrix to encode temporal information to expand search space.

Besides, we drew line charts to better reflect the NMI, Accuracy, and Purity scores of the algorithms on the four Greene datasets and four real-world datasets in Fig.3.7. The NMIs of RTSC were significantly higher than those of the baselines. For example, the NMIs of RTSC on *Cellphone* were {0.80, 0.73, 0.76, 0.78, 0.71,

Table 3.2: NMI of various algorithms on 10 temporal network datasets, where the best results among our experiments are marked in bold.

Datasets	PisCES	DYNMOGA	sE-NMF	DECS	DNE	MEGA	RTSC
SYN-FIX	0.990	0.917	0.937	**1**	0.979	0.970	**1**
SYN-VAR	0.880	0.841	0.866	0.989	0.732	0.714	**1**
Switch	0.927	0.965	0.971	0.966	0.935	0.972	**1**
Birth-Death	0.912	0.981	0.983	0.986	0.901	0.972	**0.993**
Merge-Split	0.952	0.986	0.964	0.955	0.965	0.957	**0.997**
Expand-Contract	0.926	0.973	0.978	0.974	0.951	0.967	**0.998**
Cellphone	0.593	0.704	0.608	0.717	0.671	0.682	**0.784**
Email	0.570	0.882	0.880	0.885	0.828	0.874	**0.943**
ca-cit-HepTh	0.730	0.787	0.802	0.793	0.738	0.805	**0.847**
Facebook	0.669	0.772	0.742	0.680	0.707	0.684	**0.783**

Table 3.3: Average accuracy of various algorithms on 10 temporal network datasets, where the best results among our experiments are marked in bold.

Datasets	PisCES	DYNMOGA	sE-NMF	DECS	DNE	MEGA	RTSC
SYN-FIX	0.994	0.922	0.933	**1**	0.983	0.976	**1**
SYN-VAR	0.891	0.865	0.868	0.993	0.757	0.731	**1**
Switch	0.941	0.976	0.983	0.983	0.948	0.978	**1**
Birth-Death	0.932	**0.991**	0.989	0.991	0.930	0.980	0.983
Merge-Split	0.961	0.973	0.971	0.963	0.979	0.967	**0.999**
Expand-Contract	0.939	0.981	0.986	0.982	0.962	0.976	**0.999**
Cellphone	0.620	0.734	0.641	0.734	0.659	0.715	**0.935**
Email	0.616	0.889	0.889	**0.901**	0.847	0.884	0.885
ca-cit-HepTh	0.762	0.804	0.822	0.835	0.773	0.831	**0.898**
Facebook	0.874	0.904	0.925	0.908	0.905	0.878	**0.955**

0.77, 0.74, 0.79, 0.86, 0.85$\}$ on ten snapshots, whereas those of DYNMOGA were $\{$0.68, 0.75, 0.72, 0.69, 0.69, 0.70, 0.72, 0.70, 0.71, 0.68$\}$. There are three main explanations as to why RTSC performed the best. First, jointly learning node representations and graph clustering facilitates the selection of features under the guidance of clustering, improving clustering accuracy. Second, RTSC can smooth the clustering information between successive snapshots by learning a new common embedding matrix, which can effectively eliminate noise in temporal networks. Third, RTSC adopts high-order similarity information, which is more accurate to explore the structure of temporal networks, rather than the adjacency matrix.

Table 3.4: Purity of various algorithms on 10 temporal network datasets, where the best results among our experiments are marked in bold.

Datasets	PisCES	DYNMOGA	sE-NMF	DECS	DNE	MEGA	RTSC
SYN-FIX	0.989	0.915	0.925	**1**	0.954	0.932	**1**
SYN-VAR	0.884	0.812	0.857	0.972	0.715	0.691	**1**
Switch	0.917	0.954	0.961	0.959	0.925	0.965	**1**
Birth-Death	0.901	0.970	0.974	0.979	0.888	0.954	**0.991**
Merge-Split	0.943	0.955	0.959	0.951	0.959	0.949	**0.999**
Expand-Contract	0.913	0.941	0.946	0.965	0.943	0.959	**0.999**
Cellphone	0.601	0.694	0.603	0.702	0.669	0.689	**0.947**
Email	0.572	0.863	0.861	0.869	0.832	0.847	**0.920**
ca-cit-HepTh	0.755	0.810	0.790	0.805	0.737	0.782	**0.919**
Facebook	0.879	0.915	0.933	0.877	0.867	0.874	**0.969**

3.4.2 Parameter Sensitivity

RTSC involves two hyper-parameters. α dominates the importance of the sparsity on specific noise matrices, and l denotes the dimensionality of the common embedding matrix. We selected two large-scale artificial datasets, *Switch* and *Birth-Death*, and two real-world datasets, *ca-cit-HepTh* and *Facebook*, to determine the parameter sensitivity of RTSC.

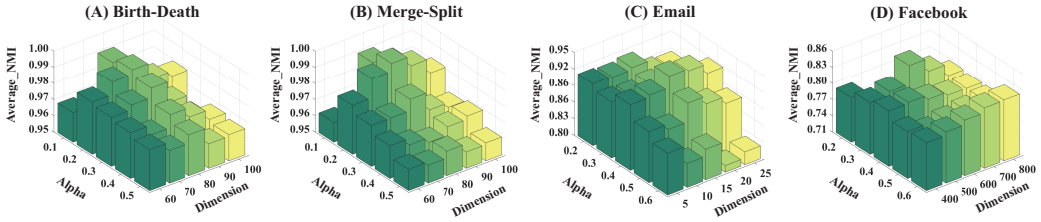

Figure 3.4: Effect of RTSC parameters on performance: (A) Birth-Death, (B) Merge-Split, (C) Email, and (D) Facebook.

We observed how α and l affect the performance of RTSC by varying α from 0.1 to 0.5 with a gap of 0.1 and l from 60 to 100 for *Switch* and *Birth-Death*, which is shown in Figs.3.4 (A)–(B). And we set α from 0.2 to 0.6 with a gap of 0.1 for *Email* and *ca-cit-HepTh*. As shown in Figs.3.4 (C)–(D), l was from 5 to 25 with a gap of 5 for *Email* and from 400 to 800 with a gap of 100 for *Facebook*. These values indicate that RTSC improves in the NMI metric when $\alpha \in [0.2, 0.4]$ and $l = 1\%N_0$, where N_0 is the total number of nodes in temporal networks.

The reason these parameters can influence the results is that when $\alpha < 0.2$,

the specific noise matrix will be condense and it becomes difficult for the common embedding matrix to extract rich shared information from successive graph embedding matrices. However, if $\alpha > 0.4$, the noise matrices may be too sparse so that great amount of noise cannot be excluded from the common graph embedding matrix, which will decrease clustering performance. If l is set to a relative small or large value, the common graph embedding matrix will contain insufficient or redundant features, thereby impairing the clustering accuracy. In order to achieve a balance in the above scenarios, we set $\alpha \in [0.2, 0.4]$ and $l = 1\%N_0$ on all datasets in our experiments.

3.4.3 Ablation Study

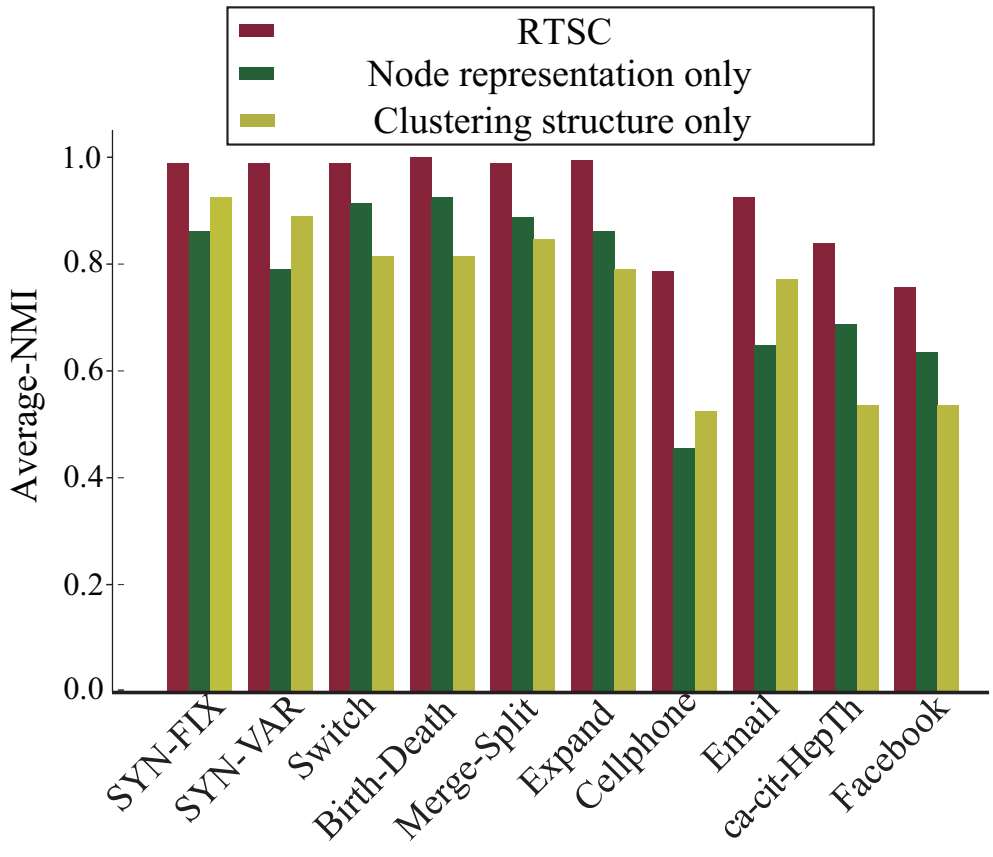

Figure 3.5: Performance comparison of RTSC with joint learning, graph-representation-learning-only, and graph-clustering-only strategies.

The above experiments indicate the superiority of RTSC. Since RTSC jointly learns node representations and graph clustering, it is natural to ask whether the

superiority of RTSC is due to either node representation learning or graph clustering structure learning. Thus, we conducted an ablation experiment by comparing RTSC with node-representation-learning-only or graph-clustering-only. The former strategy first adopts graph embeddings to learn the node representations of temporal networks and then applies K-means to obtain dynamic communities. The latter one learns a common clustering-based structure via low-rank and sparse decomposition without graph embeddings to obtain dynamic communities.

The performance of RTSC with these strategies on all datasets is shown in Fig.3.5, where the joint learning strategy significantly outperformed both of the separate learning strategies. The reason why RTSC performed best on all datasets is that it can efficiently select features from graph embeddings to construct a common embedding matrix, and this matrix can remove noise between successive time steps and effectively improve clustering accuracy. These results further indicate the superiority of RTSC in dynamic community detection in temporal networks, implying the joint learning of graph representations and clustering is promising for identifying complex dynamic communities.

3.4.4 Convergence Analysis

We then investigated the running time of RTSC, which is listed in Table 3.5, along with the running time of the four fastest baselines. RTSC was the fastest among all datasets, reducing 11.2 to 31.8% of running time. This increase in speed shows that our model can be applied to larger datasets, and it further proves the usability of RTSC in the real world.

Table 3.5: Running time of RTSC and baselines (second).

Datasets	sE-NMF	DECS	DNE	MEGA	RTSC
SYN-FIX	2.10	8.40	2.45	1.85	**1.26**
SYN-VAR	18.12	23.91	11.95	15.59	**9.22**
Switch	62,332	82,901	41,139	48,618	**33,733**
Birth-Death	59,218	76,983	38,491	47,374	**31,562**
Merge-Split	61,235	78,380	39,802	47,150	**33,433**
Expand-Con	60,343	75,428	41,636	47,067	**32,892**
Cellphone	94.23	124.38	68.32	73.47	**54.65**
Email	32.44	43.14	22.31	30.23	**19.40**
ca-cit-HepTh	1,221,323	1,477,800	830,499	964,845	**672,704**
Facebook	1,843,432	2,175,249	1,235,099	1,327,271	**988,079**

We also investigated the convergence speed of RTSC by using the relative error (normalized error using the max-min strategy). How relative error changes as the number of iterations increases is shown in Fig.3.6. We assert that the convergence speed of RTSC is much faster than traditional matrix factorization-based optimization multiplicative updates [28] and alternative least-squares [34]. RTSC generally took less than 20 iterations to converge, while the other two optimization methods required more than 200 iterations. These results indicate that RTSC can significantly accelerate the convergence speed and reduce the running time.

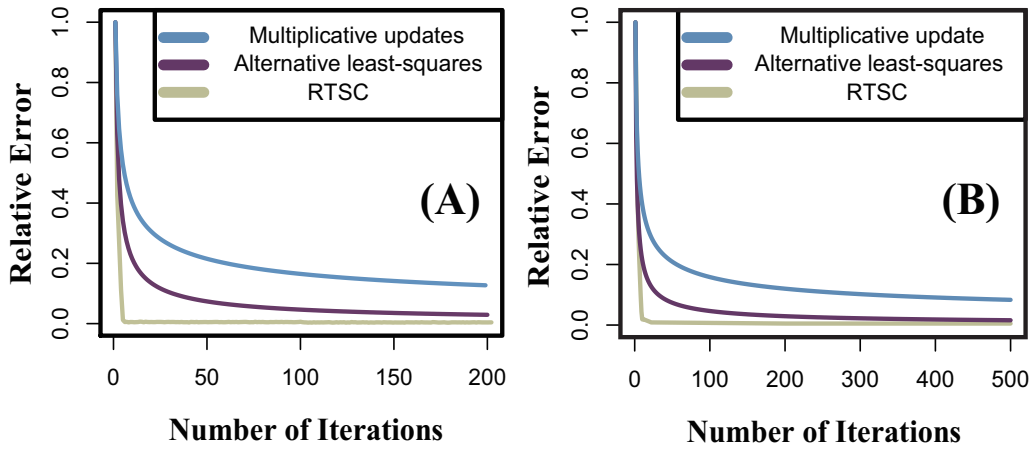

Figure 3.6: Convergence speed of RTSC: relative error vs. number of iterations on (A) Birth-Death and (B) Email.

3.5 Conclusion

In this study, we mainly focused on three challenging issues, i.e., jointly learn the node representation and graph clustering structure to improve the clustering accuracy in dynamic networks, remove noise in the smoothing procedure to enhance the gain of the common part for successive snapshots, and accelerate the convergence and reduce running time through the ALM-based optimization procedure. The experimental results on both the artificial and real-world datasets indicate that the proposed community detection algorithm significantly outperforms state-of-the-art baselines.

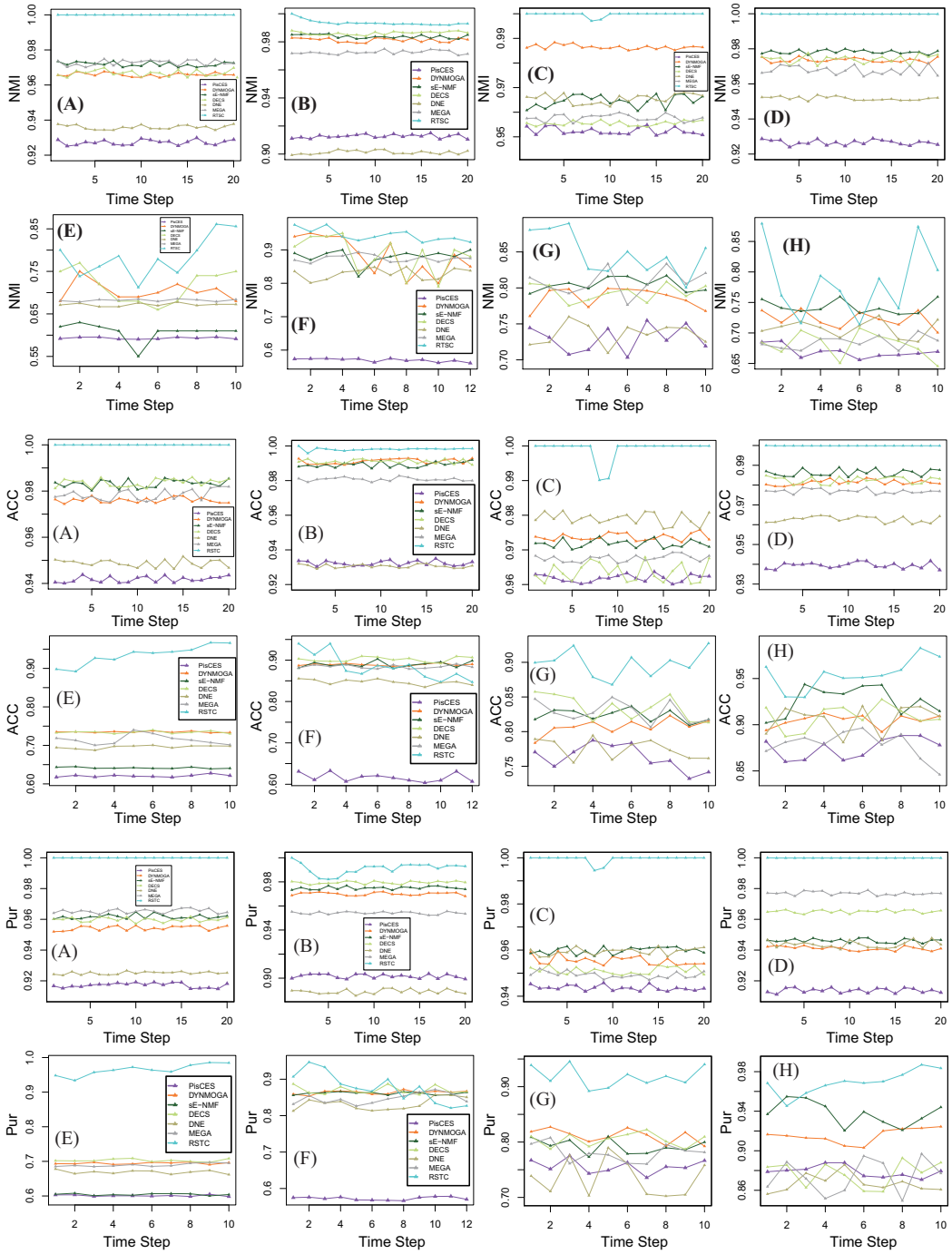

Figure 3.7: NMIs, Accuracy, and Purity for RTSC and baselines on (A) Switch, (B) Birth-Death, (C) Merge-Split, (D) Expand-Contract, (E) Cellphone, (F) Email, (G) ca-cit-HepTh, and (H) Facebook datasets respectively.

Chapter 4

Conducting joint learning on timeline summarization

4.1 Methodology

4.1.1 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.

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 meta-paths and meta-path-based neighbors for the purpose of disseminating information among heterogeneous nodes.

Definition 1 Meta-path Φ is defined as a path in the form of $v_1 \xrightarrow{e_1} \cdots \xrightarrow{e_q}$

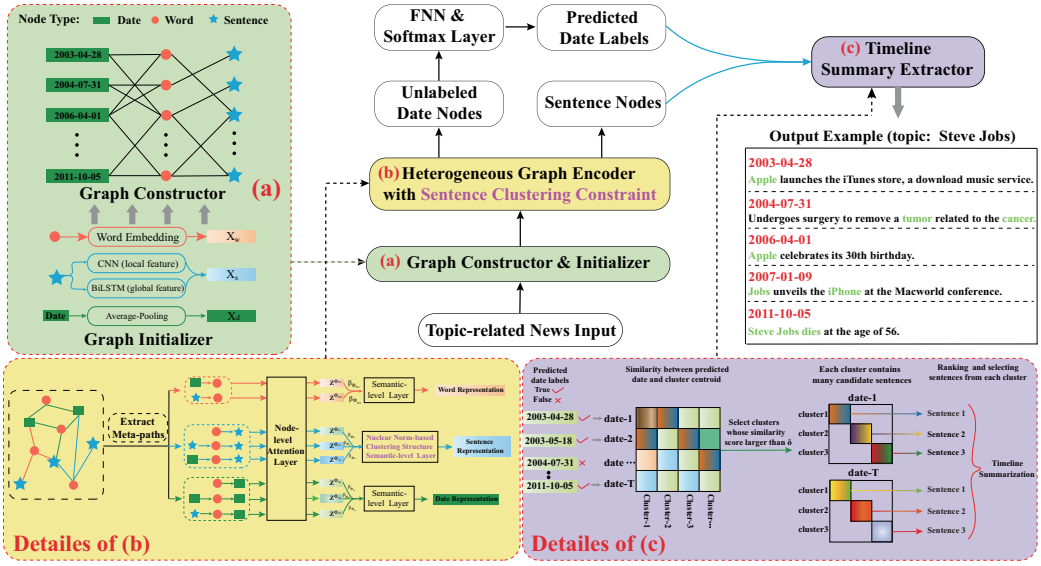

Figure 4.1: Model overview. HeterTLS consists of three chief components: (a) graph constructor and initializer, (b) heterogeneous graph encoder with sentence clustering constraint, and (c) timeline summary extractor. We first construct heterogeneous network for date, sentence, and word nodes with two 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.

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 4.1 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.

4.1.2 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 4.1 (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 convolu-

tional neural network (CNN) [35] with different kernel sizes, and q_j is gripped by a bidirectional long short-term memory (Bi-LSTM) [27]. Considering the success of transformer-based pre-trained models, we also provide another initialization strategy: using BERT [13] and sentence-BERT [61] 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 different 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 as:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{4.1}$$

$$ISF_i = \log \frac{|S|}{|\{j : w_i \in s_j\}|}, \tag{4.2}$$

$$I\text{DATE}F_i = \log \frac{|D|}{|\{j : w_i \in d_j\}|}, \tag{4.3}$$

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. 4.1 is the sum of the number of occurrences of all words in s_j or d_j. In Eqs. 4.2 and 4.3, $|S|$ and $|D|$ respectively denote the total number of sentences 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.

4.1.3 Heterogeneous graph encoder with sentence clustering constraint

As Figure 4.1 (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.

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

$$e_{ij}^{\Phi_p} = LeakyReLU(\mathbf{W}_a[\mathbf{W}_{\phi_i}\mathbf{h}_i; \mathbf{W}_{\phi_j}\mathbf{h}_j]), \tag{4.4}$$

$$\alpha_{ij}^{\Phi_p} = \frac{exp(e_{ij}^{\Phi_p})}{\sum_{l\in\mathcal{N}_i^{\Phi_p}} exp(e_{il}^{\Phi_p})}, \tag{4.5}$$

$$\mathbf{z}_i^{\Phi_p} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j\in\mathcal{N}_i^{\Phi_p}} \alpha_{ij}^{\Phi_p}\mathbf{W}_{\phi_j}\mathbf{h}_j\Big), \tag{4.6}$$

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 4.1 (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

$$w_{\Phi_p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^T tanh(\mathbf{W}\mathbf{z}_i^{\Phi_p} + \mathbf{b}), \tag{4.7}$$

$$\beta_{\Phi_p} = \frac{exp(w_{\Phi_p})}{\sum_{l=1}^{P} exp(w_{\Phi_l})}, \tag{4.8}$$

$$\mathbf{z}_i = \sum_{p=1}^{P} \beta_{\Phi_p} \mathbf{z}_i^{\Phi_p}, \tag{4.9}$$

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 [73], to avoid gradient vanishing after certain iterations, a residual connection and position-wise feed-forward network (FFN) layer with two linear transformations [71] 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

$$Z_{w_{1\sim2}}^{t+1} = NLevel(H_d^t, H_s^t, H_w^t), \tag{4.10}$$

$$H_w^{t+1} = FFN(SLevel(Z_{w_{1\sim2}}^{t+1}) + H_w^t), \tag{4.11}$$

$$Z_{d_{1\sim3}}^{t+1} = NLevel(H_d^t, H_s^t, H_w^{t+1}), \tag{4.12}$$

$$H_d^{t+1} = FFN(SLevel(Z_{d_{1\sim3}}^{t+1}) + H_d^t), \tag{4.13}$$

$$Z_{s_{1\sim3}}^{t+1} = NLevel(H_d^{t+1}, H_s^t, H_w^{t+1}), \tag{4.14}$$

$$H_s^{t+1} = FFN(SLevel(Z_{s_{1\sim3}}^{t+1}) + H_s^t), \tag{4.15}$$

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. 4.11, 4.13, and 4.15 represent the residual connection and FFN layer.

Sentence clustering constraint

Detecting main events from $\mathcal{D}$ can effectively reduce the redundancy when generating summaries. Current TLS methods [87] 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 [14]. 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 [85, 51], 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 [11] 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. [25, 58] 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

$$\mathcal{L}_{cluster} = \|H_s\|_*, \tag{4.16}$$

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 [58]. 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 4.1 (c). We also determine parameter k by the elbow method [2].

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:

$$\mathcal{L} = \mathcal{L}_{classify} + \lambda \mathcal{L}_{cluster}, \tag{4.17}$$

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. 4.17 can be optimized via stochastic gradient descent (SGD) [97] in an end-to-end manner. Readers can also refer to [58, 46] 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 4.1 (c)), thereby save much running time and improve the accuracy of TLS.

4.1.4 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 4.1 (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 4.3.4), the corresponding cluster is considered a candidate for the date. Finally, we apply CENTROID-OPT [22] as a sentence ranking algorithm within a cluster and summarize each date individually by selecting one sentence per cluster with the highest ranking score.

4.2 Experiments

4.2.1 Datasets

We carried out our experiments on the four most widely used benchmark datasets, i.e., 17 Timelines (T17) [68], Crisis [67], Entities [23], and CovidTLS [60]. 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.

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

T17 The 17 Timelines (T17) dataset [68] is the first publicly accessed and the most widely-used benchmark dataset for TLS. 17 different timeline summaries from 9 different famous topics (e.g. "BP Oil Spill", "Influenza H1N1" and "Arab Spring"), were collected from popular news agencies such as CNN, BBC, NBC-news, etc. Only those timelines with explicit timestamps (including day, month, and year) were considered. For each source articles and timeline summary pairs, the topic served as a query and time filter option to retrieve and retain top 400 news articles from the news agency that published during a specific time span. At the end, by using duplication removal, a total number of 4,650 news articles were reserved in the collection.

Crisis Crisis[1] collects event data related to long-term armed conflicts that happened in North Africa, including stories about Egypt Revolution, Syria War, Yemen Crisis, and Libya War. 25 timeline summaries acted as ground-truth, which were manually created by professional journalists and collected from the most popular news agencies, including the BBC, CNN, and Reuters. The corresponding source articles were retrieved from Google by simulating users searching for articles relevant to the timelines of the aforementioned new stories.

Entities Entities[2] contains timeline data for entities rather than events. Specifically, it consists of 47 different timelines ranging over an equal number of topics and decades of time duration. Most of the covered topics are related to life-spanning events of famous people, while the remaining ones are related to busi-

[1] http://www.l3s.de/∼gtran/timeline/

[2] https://github.com/complementizer/news-tls

ness companies and no-profit organizations. The ground-truth timelines were first obtained from CNN Fast Facts, where several hundred timelines are grouped in categories, e.g., "people" or "disasters".

CovidTLS The newly released CovidTLS dataset[3] describes the outbreak and evolution of the Covid-19 pandemic since the early 2020. As it is undoubtedly one of the most concerned worldwide events, it has been reported by an unprecedented amount of news articles. The source articles were crawled from several English-written journals, and the ground-truth timeline was retrieved from a public, authoritative website. With peculiar characteristics, the whole dataset has just one, complex topic (i.e., the outbreak Covid-19 pandemic) reported by over 26 thousands news articles. The number of candidate dates in CovidTLS is one order of magnitude higher than those of all the existing topics.

Using these datasets makes it possible to comprehensively verify the effectiveness and generalization of HeterTLS because both the number of topics and their time spans among them are completely different. Specifically, the Entities dataset contains dozens of topics and spans decades of news articles per topic, while the others involve only a few topics within two years. The basic statistics of the datasets, including the splitting details, are summarized in Table 4.1.

4.2.2 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

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

reasonable for news articles. News, by definition, reports up-to-date events.

4.2.3 Evaluation metrics

In our experiments, the evaluation of model-generated timelines depended on the ROUGE metric and its variants as follows [87]:

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 [83, 50, 76].

Alignment-based ROUGE F1 On the basis of the above concatenation metric, it linearly penalizes the ROUGE score by the distance of date alignments [48].

Date selection F1 It only measures how well the model selects dates contained in the ground-truth [49].

4.2.4 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. 4.3.5. All experiments for a dataset were subject to leave-one-out cross-validation, and significant differences were determined by bootstrap test [15] 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 [56]. 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.

Datasets	T17					Crisis				
Metrics	CR1-F	CR2-F	AR1-F	AR2-F	Date-F1	CR1-F	CR2-F	AR1-F	AR2-F	Date-F1
Full Oracle	0.500	0.180	0.312	0.128	0.926	0.490	0.160	0.360	0.150	0.974
CHIEU (2004)	0.290	0.072	0.067	0.019	0.252	0.374	0.070	0.052	0.012	0.142
TRAN (2013)	0.336	0.065	0.094	0.022	0.517	0.271	0.034	0.054	0.012	0.289
MARTS. (2018)	0.383	0.092	0.105	0.030	0.544	0.333	0.072	0.075	0.016	0.281
DATEWISE (2020)	0.385	0.097	0.121	0.035	0.544	0.347	0.075	0.089	0.026	0.295
DASG (2021)	0.333	0.064	0.118	0.029	0.647	0.323	0.068	0.077	0.018	0.381
SDF (2021)	0.401	0.101	0.106	0.033	0.553	0.360	0.073	0.064	0.014	0.302
HeterTLS-HAN	0.398	0.101	0.141	0.052	0.668	0.372	0.070	0.092	0.026	0.455
HeterTLS-Joint	0.392	0.101	0.132	0.042	0.620	0.323	0.068	0.079	0.015	0.418
HeterTLS+Pre	0.401	0.103	0.142	0.053	0.688	**0.379**†	**0.078**†	**0.107**†	**0.028**†	**0.494**†
HeterTLS	**0.408**†	**0.108**†	**0.145**†	**0.058**†	**0.703**†	0.374	0.075	0.105	0.028	0.492

Table 4.2: Concatenation- and alignment-based ROUGE-1/2 F1-scores for T17 and Crisis datasets. Best results among model-generated timelines are marked in bold. Symbol † indicates that our results significantly surpass all baselines using bootstrap test [15] with $p < 0.005$.

4.2.5 Baselines

To interpret the concatenation- and alignment-based ROUGE scores better and to approximate their upper bounds, we measure the performance of the full oracle method:

- FULL ORACLE: Selects the correct dates and constructs a summary for each date by optimizing the ROUGE to the ground-truth summaries.

The following excellent baselines were used for comparison and to demonstrate the effectiveness of HeterTLS: *direct summarization* including CHIEU [10] and MARTSCHAT [49]; *date-wise summarization* such as TRAN [68], DATEWISE [23], and SDF [60]; and *event detection* method DASG [44]. We additionally follow [23] to obtain full oracle.

4.3 Results and Discussion

4.3.1 Performance of HeterTLS

According to Tables 4.2 and 4.3, HeterTLS outperformed all baselines in terms of all metrics. Considering that DASG ignores date information, we excluded it

Datasets	Entities					CovidTLS				
Metrics	CR1-F	CR2-F	AR1-F	AR2-F	Date-F1	CR1-F	CR2-F	AR1-F	AR2-F	Date-F1
Full Oracle	0.348	0.079	0.232	0.075	0.757	0.471	0.199	0.388	0.192	0.968
CHIEU (2004)	0.275	0.053	0.036	0.011	0.102	0.203	0.021	0.008	0.001	0.176
TRAN (2013)	0.275	0.052	0.042	0.012	0.185	0.218	0.028	0.012	0.001	0.675
MARTS. (2018)	0.275	0.052	0.042	0.011	0.167	0.249	0.036	0.028	0.001	0.685
DATEWISE (2020)	0.271	0.051	0.057	0.017	0.205	0.318	0.038	0.036	0.005	0.697
DASG (2021)	0.282	0.052	0.045	0.010	0.372	0.224	0.030	0.014	0.001	0.621
SDF (2021)	0.275	0.052	0.041	0.011	0.397	0.439	0.076	0.062	0.011	0.689
HeterTLS-HAN	0.272	0.052	0.054	0.015	0.432	0.402	0.062	0.052	0.009	0.656
HeterTLS-Joint	0.271	0.048	0.049	0.012	0.395	0.388	0.058	0.048	0.006	0.648
HeterTLS+Pre	0.282	0.054	0.057	0.019	0.478	**0.447**†	**0.078**†	**0.068**†	**0.012**†	**0.722**†
HeterTLS	**0.288**†	**0.058**†	**0.059**†	**0.019**†	**0.488**†	0.430	0.072	0.060	0.011	0.704

Table 4.3: Concatenation- and alignment-based ROUGE-1/2 F1-scores for Entities and CovidTLS datasets. Best results among model-generated timelines are marked in bold. Symbol † indicates that our results significantly surpass all baselines using bootstrap test [15] with $p < 0.005$.

from the Date F1 experiment. We noticed that HeterTLS with pre-trained initial node representations surpassed HeterTLS only on Crisis and CovidTLS 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.

Theoretically, it should achieve a Date-F1 score of 100%. In practice, its Date-F1 scores end up being lower due to the fact that, for certain dates, no candidate sentences matching the query key phrases can be located. This leads to the exclu-

sion of those dates from the oracle timelines.

4.3.2 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 4.2 and 4.3 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 [73]. 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 4.2 and 4.3, 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 the combination of node representations and clustering structure is promising for identifying salient dates and sentence candidates simultaneously.

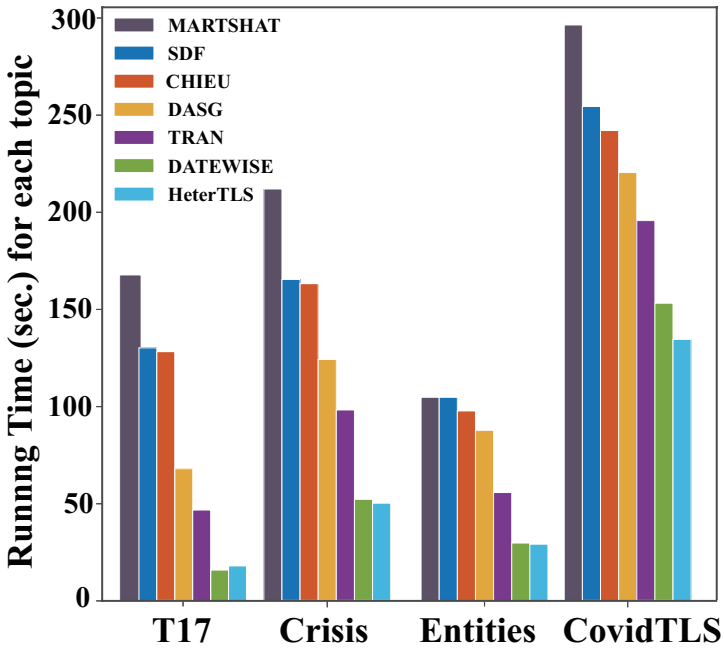

Figure 4.2: Comparison of running time of current state-of-the-art models and HeterTLS

4.3.3 Running time

We conducted an investigation of running time with all models being trained with the same device and show the results in Figure 4.2. 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 [83]. 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.

4.3.4 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. 4.17, and δ acts as a threshold to decide the most related clusters for selected dates (Figure 4.1(c)). We selected several sets of λ and δ to test the performance of HeterTLS in terms of AR1-F and AR2-F and give a general overview in Figure 4.3. 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, Entities, and CovidTLS datasets. It is explicit that a larger δ coupled with a smaller λ 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. Therefore, we reasonably believe that our proposed model is not sensitive to parameters.

4.3.5 Ratio of labeled dates

Figure 4.4 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.

4.3.6 Consecutive dates and redundancy

The proportions of consecutive dates in chronologically ordered model-generated timelines and ground-truth timelines were experimentally measured according to [23]. 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.

Combining with Table 4.1, Table 4.4 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

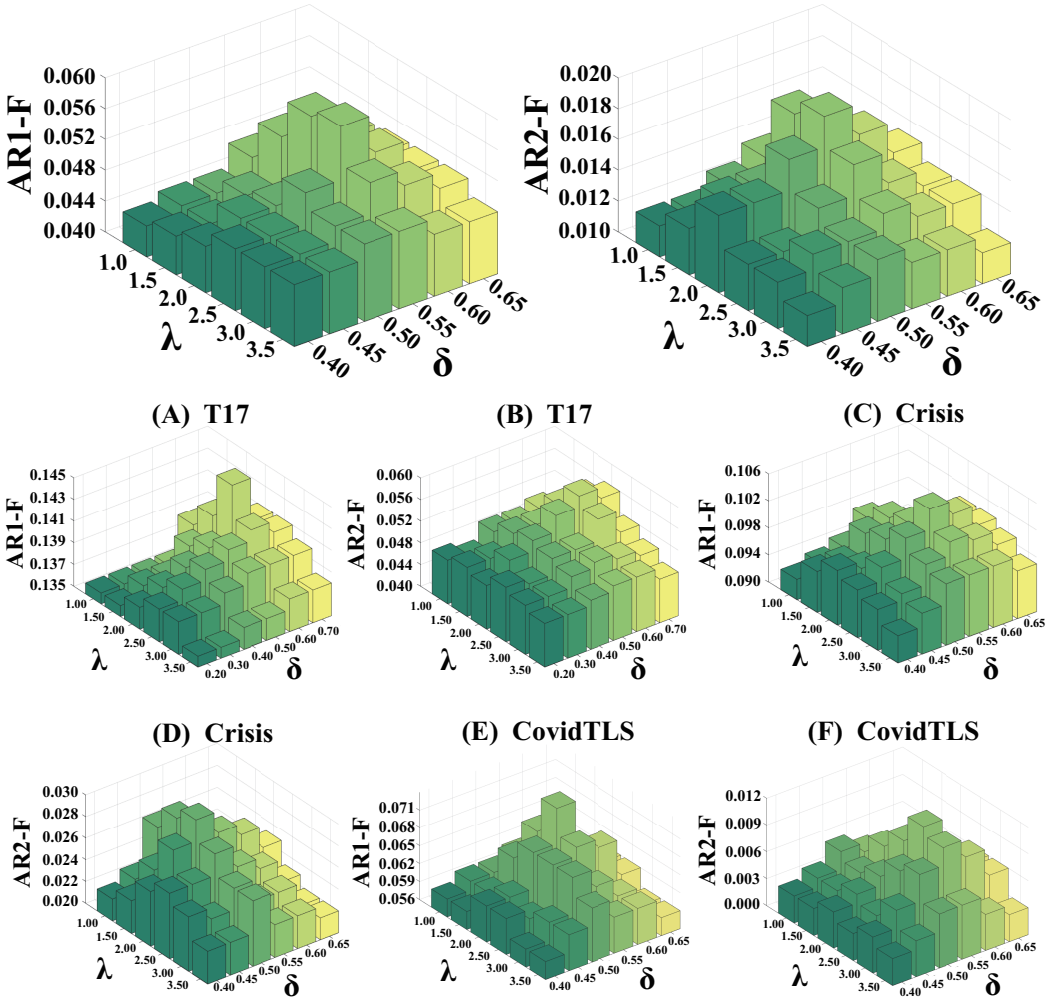

Figure 4.3: Impact of parameters on T17, Crisis, Entities, and CovidTLS dataset

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

adjacent date proportion is the same as that of redundancy. The results in Table 4.4 indicate that HeterTLS is the closest to the ground-truth, thereby proving its ability to predict salient dates.

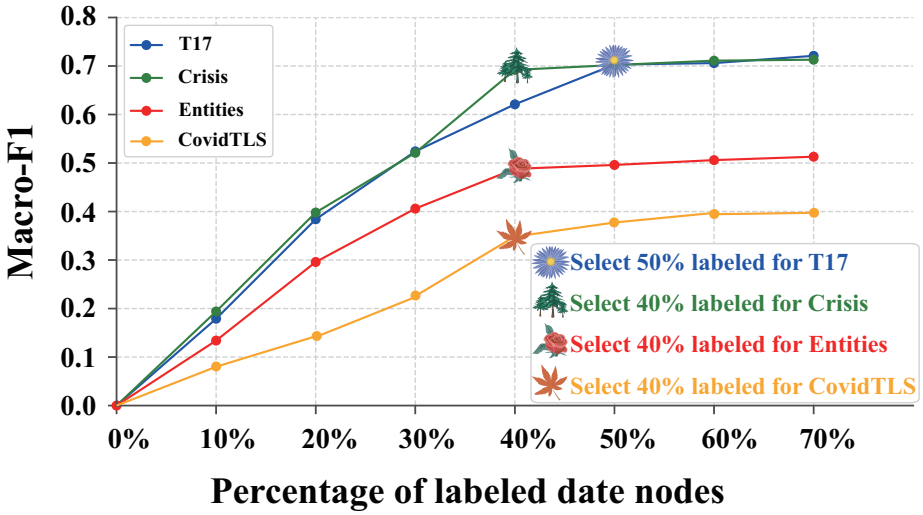

Figure 4.4: Ratio of labeled date nodes on training set vs. corresponding accuracy on test set

4.3.7 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 4.5, parts of the ground-truth timeline of certain dates are shown on the left, while Figure 4.6 lists the HeterTLS-generated timeline with similar period coverage as the ground-truth. We manually colored some keywords to illustrate consistent content in both timeline summaries. Readers can concretely judge our model by precise dates (marked in red) and subjects (marked in orange). 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 sum-

mary, 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.

4.4 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.

Entities Dataset

Topic: Steve_Jobs Ground-truth timeline

2003-04-28

Apple launches the iTunes store, a download music service.

2004-07-31

Undergoes surgery to remove a tumor related to the cancer.

2006-04-01

Apple celebrates its 30th birthday.

2007-01-09

Jobs unveils the iPhone at the Macworld conference.

2008-06-27

A class action suit is filed against Jobs and several members of the Apple's board of directors, claiming that they had participated in the backdating of stock option grants. In 2006, Apple was forced to restate its financial results after acknowledging that an internal investigation had revealed irregularities in its stock option grants between 1997 and 2001.

2008--2009 ● ● ● ● ● ●

2009-06-29

Apple spokesman Steve Dowling announces that Jobs has returned to work.

2010-01-27

Jobs introduces the iPad. The half-inch-thick, 1.5pound 9.7inch iPad allows users to read books, play games or watch video.

2011-03-02

Jobs receives a standing ovation when he takes the stage to unveil the iPad 2.

2011-06-06

At the Worldwide Developers Conference (WWDC) Jobs introduces iCloud the new online media storage system. Other Apple officials demo the new operating systems OS-X Lion and iOS-5.

2011-08-24

Resigns as CEO of Apple, but announces he will stay on as chairman. Tim Cook is promoted to CEO.

Figure 4.5: Parts of a ground-truth timeline summary on the topic of Steve Jobs.

Topic: Steve_Jobs HeterTLS-generated timeline

2003-04-28

Apple launches the Powerbook laptop.

2004-07-31

When Jobs was recovering from surgery to remove the original cancer.

2006-04-01

Appropriately enough, April 1 is the date Apple plans to celebrate its 30th birthday.

2008-06-27

The computer maker said it has brought in independent counsel to review the handing out of options between 1997 and 2001, including a batch for chief executive Steve Jobs, after an internal inquiry found potential irregularities, Steve Jobs cooperated with Apple 's independent investigation and with the government 's investigation of stock option grants at Apple, directors said.

2008--2009 ● ● ● ● ● ●

2009-06-29

Apple spokesman Steve Dowling said the decision to pull Jobs out of the show indicated the company 's intention to stop exhibiting at Macworld . Big Brother permitting the choice that Jobs has made this time round.

2010-01-27

Apple says that it “lets users browse the web, read and send email, enjoy and share photos, watch videos, listen to music, play games, read ebooks and much more on the Mac, the iPad, the iPod , the iPhone”.

2011-03-02

We should know in a couple of hours, when Jobs takes the stage to keynote (sic) Apple 's Worldwide Developer Conference 2005, which opens today in San Francisco. A rapturous standing ovation follows.

2011-06-06

Next week is the Apple Worldwide Developers' Conference, where Steve Jobs will address the adoring masses. Details of the new operating systems have been dribbling out for months, with the official unveiling in October last year.

2011-08-24

Tim Cook is widely tipped as a possible replacement as CEO. I hereby resign as CEO of Apple.

Figure 4.6: Parts of a model-generated timeline summary on the topic of Steve Jobs produced by HeterTLS.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Temporal networks and time series article collections provide an effective means to characterize the dynamics of complex systems and the development of events. Dynamic community detection and timeline summarization are promising to reveal the complicated mechanisms of such systems. However, compared with static community detection and extractive summarization, less attention has been paid to dynamic community detection and timeline summarization, which leads to many problems remaining.

Temporal networks provide an effective means to characterize the dynamics of complex systems. However, compared with static community detection, less attention has been paid to dynamic community detection, which leads to many problems remaining. In our first study, we mainly focused on three challenging issues, i.e., jointly learn the node representation and graph clustering structure to improve the clustering accuracy in dynamic networks, remove noise in the smoothing procedure to enhance the gain of the common part for successive snapshots, and accelerate the convergence and reduce running time through the ALM-based optimization procedure. The experimental results on both the artificial and real-world datasets indicate that the proposed community detection algorithm significantly outperforms state-of-the-art baselines.

In the second work, we have compared and proposed different strategies to construct timeline summaries of long-ranging news topics: the previous state-of-the-art method based on direct summarization, a date-wise approach, and a clustering-

based approach. 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.

5.2 Future Work

Finally, we present some directions for future work as follows:

In our first proposal, choosing three snapshots to extract common parts may be a difficult issue, as we may need to consider the dynamics of the graph at different time scales. In the future, a more flexible approach could be considered to select or merge multiple snapshots at various times. Besides, we also found that the nuclear norm may sometimes lead to over-sparsification, i.e., the extracted common parts become too sparse, losing some important structural information. For future work, on one hand, improvements could be considered for these potential problems, such as adopting more efficient algorithms to reduce computational complexity or adding regularization terms to control sparsity. On the other hand, exploring the combination of these methods with other types of information (like node attributes or types of edges) could be considered to further improve the performance of dynamic graph clustering.

In our second proposal, the method depends on the calculated similarity between the date representation and sentence representation. However, a high similarity score does not necessarily guarantee that the sentence provides useful or diverse information for the summary. In the future, the introduction of additional criteria for sentence selection, such as relevance to the overall context or diversity with comparison to already selected sentences, might improve the quality of the summary. Furthermore, text data is often noisy, and incorrect sentence representations can significantly affect the quality of the summary. A robust model that can effectively deal with noise, perhaps by introducing attention mechanisms or data-cleaning techniques, could be beneficial. For future work, one could look into

improving the model's robustness to noise, incorporating additional criteria for sentence selection, reducing the computational complexity, and enhancing the model's interpretability. Furthermore, the method could be expanded to incorporate other relevant features such as sentence sentiment, entity recognition, or topic modeling to better capture the context and content of the sentences.

All in all, constructing data into a graph can offer various benefits, especially when dealing with structured or relational data. Graphs are a powerful representation that can capture complex relationships between entities in a more intuitive and efficient manner than traditional tabular or sequential data structures. They have been successful in various applications, including node classification, link prediction, and even image captioning. Especially in image captioning, an image can be treated as a graph where nodes represent regions or objects within the image, and edges capture spatial relationships between these regions. This graph representation can capture important contextual information. Therefore, we believe that the field of graph deserves more in-depth research and application.

References

[1] James Allan, Rahul Gupta, and Vikas Khandelwal. 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, 2001.

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

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[4] Zhan Bu, Huijia Li, Chengcui Zhang, Jie Cao, Aihua Li, and Yong Shi. Graph k-means based on leader identification, dynamic game, and opinion dynamics. *IEEE Transactions on Knowledge and Data Engineering*, 32(7):1348–1361, 2019.

[5] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.

[6] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Geometry interaction knowledge graph embeddings. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022,*

The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 5521–5529. AAAI Press, 2022.

[7] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 554–560, Philadelphia, PA, USA, 2006. ACM.

[8] Lu Chen, Chengfei Liu, Rui Zhou, Jiajie Xu, Jeffrey Xu Yu, and Jianxin Li. Finding effective geo-social group for impromptu activities with diverse demands. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 698–708, Virtual Event, CA, USA, 2020. ACM.

[9] Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 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, 2019.

[10] Hai Leong Chieu and Yoong Keok Lee. 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, 2004.

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

[12] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of statistical mechanics: Theory and experiment*, 2005(09):P09008, 2005.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, 2019.

[14] Chris H. Q. Ding and Tao Li. Adaptive dimension reduction using discriminant analysis and K-means clustering. In Zoubin Ghahramani, editor, *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, 2007.

[15] Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 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, 2018.

[16] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. Dynamic network embedding : An extended approach for skip-gram based network embedding. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2086–2092, Stockholm, Sweden, 2018. ijcai.org.

[17] Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. 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, 2020.

[18] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference*

(ICML 2008), Helsinki, Finland, June 5-9, 2008, volume 307 of *ACM International Conference Proceeding Series*, pages 272–279, Helsinki, Finland, 2008. ACM.

[19] Francesco Folino and Clara Pizzuti. An evolutionary multiobjective approach for community discovery in dynamic networks. *IEEE Trans. Knowl. Data Eng.*, 26(8):1838–1852, 2014.

[20] Dongqi Fu, Dawei Zhou, and Jingrui He. Local motif clustering o time-evolving graphs. In *Conference on Knowledge Discovery and Data Mining*, pages 390–400, CA, USA, 2020. ACM.

[21] Michael Gao, Lindsay Popowski, and Jim Boerkoel. Dynamic control of probabilistic simple temporal networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 9851–9858, New York, NY, USA, 2020. AAAI Press.

[22] Demian Gholipour Ghalandari. 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, 2017.

[23] Demian Gholipour Ghalandari and Georgiana Ifrim. 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, 2020.

[24] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864, CA, USA, 2016. ACM.

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

[26] Dongxiao He, Yue Song, Di Jin, Zhiyong Feng, Binbin Zhang, Zhizhi Yu, and Weixiong Zhang. Community-centric graph convolutional network for unsupervised community detection. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3515–3521, Yokohama, Japan, 2020. ijcai.org.

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

[28] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, 2004.

[29] Yun Hu, Yeshuang Zhu, Jinchao Zhang, Changwen Zheng, and Jie Zhou. 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, 2021.

[30] Ling Huang, Hong-Yang Chao, and Guangqiang Xie. Mumod: A micro-unit connection approach for hybrid-order community detection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 107–114, New York, NY, 2020. AAAI Press.

[31] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 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, 2020.

[32] Di Jin, Kunzeng Wang, Ge Zhang, Pengfei Jiao, Dongxiao He, Francoise Fogelman-Soulie, and Xin Huang. Detecting communities with multiplex semantics by distinguishing background, general, and specialized topics. *IEEE Transactions on Knowledge and Data Engineering*, 32(11):2144–2158, 2019.

[33] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2(1):622–633, 2009.

[34] Keigo Kimura, Yuzuru Tanaka, and Mineichi Kudo. A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization. In Dinh Q. Phung and Hang Li, editors, *Proceedings of the Sixth Asian Conference on Machine Learning, ACML 2014, Nha Trang City, Vietnam, November 26-28, 2014*, volume 39 of *JMLR Workshop and Conference Proceedings*, pages 129–141, Nha Trang City, Vietnam, 2014. JMLR.org.

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

[36] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[37] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185, Montreal, Quebec, Canada, 2014. NeurIPS.

[38] Jiwei Li and Sujian Li. 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, 2013.

[39] Xi Li, Qianren Mao, Hao Peng, Hongdong Zhu, Jianxin Li, and Zheng Wang. Automated timeline length selection for flexible timeline summarization. *CoRR*, abs/2105.14201, 2021.

[40] Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. Community detection in attributed graphs: An embedding approach. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 338–345, New Orleans, Louisiana, USA, 2018. AAAI Press.

[41] Fan Liu and Yong Deng. Determine the number of unknown targets in open world based on elbow method. *IEEE Trans. Fuzzy Syst.*, 29(5):986–995, 2021.

[42] Fanzhen Liu, Jia Wu, Shan Xue, Chuan Zhou, Jian Yang, and Quanzheng Sheng. Detecting the evolving community structure in dynamic social networks. *World Wide Web*, 23(2):715–733, 2020.

[43] Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences*, 115(5):927–932, 2018.

[44] Jingzhou Liu, Dominic J. D. Hughes, and Yiming Yang. 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, 2021.

[45] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7603–7611. AAAI Press, 2022.

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

[47] Xiaoke Ma and Di Dong. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(5):1045–1058, 2017.

[48] Sebastian Martschat and Katja Markert. 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, 2017.

[49] Sebastian Martschat and Katja Markert. 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, 2018.

[50] Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 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, 2014.

[51] Feiping Nie, Xiaoqian Wang, Cheng Deng, and Heng Huang. Learning A structured optimal bipartite graph for co-clustering. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *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, Long Beach, CA, USA, 2017. NeurIPS.

[52] Feiping Nie, Han Zhang, Rong Wang, and Xuelong Li. Semi-supervised clustering via pairwise constrained optimal graph. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3160–3166, Yokohama, Japan, 2020. ijcai.org.

[53] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

[54] Arian Pasquali, Vítor Mangaravite, Ricardo Campos, Alípio Mário Jorge, and Adam Jatowt. 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, 2019.

[55] Leto Peel and Aaron Clauset. Detecting change points in the large-scale structure of evolving networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2914–2920, Austin, TexasUSA, 2015. AAAI Press.

[56] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *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, 2014.

[57] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710, New York, NY, USA, 2014. ACM.

[58] Xinglin Piao, Yongli Hu, Junbin Gao, Yanfeng Sun, and Baocai Yin. 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, 2019.

[59] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek, editors, *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 459–467, Marina Del Rey, CA, USA, 2018. ACM.

[60] Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. Summarize dates first: A paradigm shift in timeline summarization. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *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, 2021.

[61] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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, 2019.

[62] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

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

[64] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 687–696, California, USA, 2007. ACM.

[65] Satoko Suzuki and Ichiro Kobayashi. 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, 2014.

[66] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1067–1077, Florence, Italy, 2015. ACM.

[67] Giang Binh Tran, Mohammad Alrifai, and Eelco Herder. 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, 2015.

[68] Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 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, 2013.

[69] Giang Binh Tran, Eelco Herder, and Katja Markert. 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, 2015.

[70] Tuan Tran, Claudia Niederée, Nattiya Kanhabua, Ujwal Gadiraju, and Avishek Anand. 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, 2015.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 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, 2017.

[72] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[73] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 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, 2020.

[74] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. Graph learning based recommender systems: A review. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4644–4652. ijcai.org, 2021.

[75] Wenjing Wang and Xiang Li. Temporal stable community in time-varying networks. *IEEE Trans. Netw. Sci. Eng.*, 7(3):1508–1520, 2020.

[76] William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. 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, 2016.

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

[78] Zhixiao Wang, Zechao Li, Guan Yuan, Yunlian Sun, Xiaobin Rui, and Xinguang Xiang. Tracking the evolution of overlapping communities in dynamic social networks. *Knowl. Based Syst.*, 157:81–97, 2018.

[79] Haozhe Wu, Zhiyuan Hu, Jia Jia, Yaohua Bu, Xiangnan He, and Tat-Seng Chua. Mining unfollow behavior in large-scale online social networks via spatial-temporal interaction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 254–261, New York, NY, USA, 2020. AAAI.

[80] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 2149–2155, Québec City, Québec, Canada, 2014. AAAI Press.

[81] Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 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, 2021.

[82] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 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, 2011.

[83] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 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, 2011.

[84] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang. Modularity based community detection with deep learning. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2252–2258, New York, NY, USA, 2016. IJCAI/AAAI Press.

[85] Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. Robust dynamic clustering for temporal networks. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2424–2433. ACM, 2021.

[86] Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. Abstractive document summarization with word embedding reconstruction. In Galia Angelova, Maria Kunilovskaya, Ruslan Mitkov, and Ivelina Nikolova-Koleva, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3September, 2021*, pages 1586–1596, Online, 2021. INCOMA Ltd.

[87] Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 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, 2021.

[88] Xiangxiang Zeng, Wen Wang, Cong Chen, and Gary G. Yen. A consensus community-based particle swarm optimization for dynamic community detection. *IEEE Trans. Cybern.*, 50(6):2502–2513, 2020.

[89] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Trans. Big Data*, 6(1):3–28, 2020.

[90] Jianlei Zhang, Yuying Zhu, and Zengqiang Chen. Evolutionary game dynamics of multiagent systems on multiple community networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11):4513–4529, 2020.

[91] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Trans. Image Process.*, 29:5491–5506, 2020.

[92] Wayne Xin Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 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, 2013.

[93] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Mach. Learn.*, 55(3):311–331, 2004.

[94] Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. 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, 2021.

[95] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. Scalable temporal latent space inference for link prediction in dynamic social networks. In *33rd IEEE International Conference on Data Engineering*, pages 57–58, San Diego, CA, USA, 2017. IEEE Computer Society.

[96] Di Zhuang, J. Morris Chang, and Mingchen Li. Dynamo: Dynamic community detection by incrementally maximizing modularity. *IEEE Trans. Knowl. Data Eng.*, 33(5):1934–1945, 2021.

[97] Martin Zinkevich, Markus Weimer, Alexander J. Smola, and Lihong Li. Parallelized stochastic gradient descent. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *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., 2010.

Acknowledgement

First and foremost, I am extremely grateful to my supervisor, Prof. Manabu Okumura, who has provided me with valuable guidance and support during the whole period of my Ph.D. course at Tokyo Institute of Technology. His kindness and vigorous academic ability encourage me not only in my research but also in my future study. Professor Okumura also gave me enough space in research so that I can take my time to explore and choose the research I am interested in. It ' s my honor to be Professor Okumura ' s student.

All thanks to the members of Okumura Laboratory. Many people in this laboratory gave me much help in my study and life, especially, Associate Professor Hidetaka Kamigaito. I often discuss research details and ideas with Associate Professor Kamigaito, and he taught me a lot with great patience. Chenlong Hu and Dongyuan Li also gave me many helpful suggestions for my research. Special thanks to the administrative assistant Iiyama-san, who helped me a lot with English proofreading for papers, journals, and academic conferences.

I would also like to thank the anonymous reviewers of the international conferences/journal for their valuable comments and suggestions that further improved my work. I am also grateful to all the other members of my thesis committee, Prof. Itsuo Kumazawa, Minoru Nakayama, Takahiro Shinozaki, and Kotaro Funakoshi for their insightful comments and advice on this thesis.

Finally, I will not forget to appreciate my family and friends for their support, caring, and help. I dedicate my thesis to all of them.

List of Publications

Journal Papers Related to This Thesis

- Jingyi You, Dongyuan Li, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura. Joint Learning-based Heterogeneous Graph Attention Network for Timeline Summarization. *Journal of Natural Language Processing, 2023, 30(1): 184-214*.

Conference Papers Related to This Thesis

- Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura. Robust Dynamic Clustering for Temporal Networks. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*.

- Jingyi You, Dongyuan Li, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura. Joint Learning-based Heterogeneous Graph Attention Network for Timeline Summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2022*.

Other Conference Papers

- Jingyi You, Dongyuan Li, Manabu Okumura, Kenji Suzuki. JPG - Jointly Learn to Align: Automated Disease Prediction and Radiology Report Generation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*.

- Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Hiroya Takamura, Manabu Okumura. Abstractive Document Summarization with Word Embedding Reconstruction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*.

- Dongyuan Li, Jingyi You, Kotaro Funakoshi, Manabu Okumura. A-TIP: Attribute-aware Text Infilling via Pre-trained Language Model. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*.

- Tatsuya Ishigaki, Jingyi You, Hiroki Takimoto, Manabu Okumura. Supporting Information Recall for Elderly People in Hyper Aged Societies. HCI (28) 2020: 282-291 *Human Aspects of IT for the Aged Population. Healthy and Active Aging - 6th International Conference, ITAP 2020, Held as Part of the 22nd HCI International Conference, HCII 2020*.