World Wide Web
https://doi.org/10.1007/s11280-017-0501-x

# Event phase oriented news summarization

**Chengyu Wang[1] · Xiaofeng He[1] · Aoying Zhou[2]**

Received: 12 January 2017 / Revised: 18 September 2017 / Accepted: 29 September 2017


**Abstract** Event summarization is a task to generate a single, concise textual representation of an event. This task does not consider multiple development phases in an event. However, news articles related to long and complicated events often involve multiple phases. Thus, traditional approaches for event summarization generally have difficulty in capturing event phases in summarization effectively. In this paper, we define the task of Event Phase Oriented News Summarization (EPONS). In this approach, we assume that a summary contains multiple timelines, each corresponding to an event phase. We model the semantic relations of news articles via a graph model called Temporal Content Coherence Graph. A structural clustering algorithm EPCluster is designed to separate news articles into several groups corresponding to event phases. We apply a vertex-reinforced random walk to rank news articles. The ranking results are further used to create timelines. Extensive experiments conducted on multiple datasets show the effectiveness of our approach.

**Keywords** Event phase · News summarization · Structural clustering · Timeline generation · Vertex-reinforced random walk

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering*
Guest Editors: Wojciech Cellary, Hua Wang, and Yanchun Zhang

✉ Xiaofeng He
xfhe@sei.ecnu.edu.cn

Chengyu Wang
chywang2013@gmail.com

Aoying Zhou
ayzhou@sei.ecnu.edu.cn

[1] Shanghai Key Laboratory of Trustworthy Computing, School of Computer Science and Software Engineering, East China Normal University, Shanghai, 200062, China

[2] School of Data Science and Engineering, East China Normal University, Shanghai, 200062, China

## 1 Introduction

Due to the overwhelming amount of information on the Web, it is time consuming for readers to digest the information. Hence, it is very important to generate summaries from massive news articles automatically [11, 17, 20, 34]. The task is to generate a single and concrete textual representation of the event. The accuracy and conciseness of these event summaries are essential for Web-based applications, such as Web search [14], news recommendation [1], semantic retrieval [15], etc.

Most existing summarization approaches treat the event as a single phase and generate a single summary. These methods ignore the existence of different phases in a certain event. We take Egypt Revolution as example. According to the Wikipedia page, major phases of this revolution include Protests against Hosni Mubarak, Egypt under the Supreme Council of the Armed Forces, Egypt under President Mohamed Morsi, June 2013 Protests against President Morsi, etc.[1] Summarizing these event phases into a single document makes it difficult for readers to comprehend how the event was evolved. Recently, inspired by the task of timeline generation, research focuses on time-relevant component summaries for an event [38, 47]. However, we notice that summaries gathered in a single timeline are simply ordered by time, which can be incomplete and less-structured for complicated events.

In order to provide better analysis for events in news articles, we introduce the task of Event Phase Oriented News Summarization (EPONS). The goal of this task is to cluster news articles from a news corpus automatically, based on event phases, and generate multiple summaries corresponding to different event phases. In this paper, an event phase summary is represented as a timeline, describing the development trend of the event phase. This task is interesting and novel for the following reasons: (i) it considers the inner features of news articles and provides a new perspective (i.e., event phases) to cluster accordingly, instead of using content similarity or publication time only; (ii) it helps readers understand complicated events more clearly and deeply, with the help of event phase summaries; and (iii) it can further improve the performance of tasks like document summarization and timeline generation.

In this paper, a "divide-and-conquer" strategy is applied to generate timelines for each event phase, after clustering news articles into event phases. We turn the event phases expressed in natural language text into two underlying semantic relations (i.e., content coherence and temporal influence) by a graph-based model, called Temporal Content Coherence Graph (TCCG). We then apply EPCluster, a structural clustering algorithm to group news articles. After this process, each phase corresponds to a generated cluster of news articles. For news articles related to a single event phase, we design a ranking algorithm based on vertex-reinforced random walk to calculate the relevance scores of news articles and the quality of news headlines. Similar to previous research [38], we define a timeline as top-$k$ news headlines and their publication time. For each timeline, an approximate optimization algorithm is employed to select news articles, in a greedy manner.

This paper is an improved version of the conference paper entitled "Event Phase Extraction and Summarization" presented in WISE 2016 [44]. We extend the conference paper in the following major aspects. (i) We present a more detailed discussion of related work and our approach. (ii) A new headline quality ranking model is proposed as part of the timeline generation method. (iii) We conduct new experiments and case studies, and present a deeper study to analyze the experimental results.

[1] See background info at: https://en.wikipedia.org/wiki/Egyptian_Revolution_of_2011.

In summary, three major contributions in this paper are concluded as follows:

- A graph structure TCCG is introduced to model the semantic relations among news articles based on content coherence and temporal influence.
- We propose a structural clustering algorithm called EPCluster to gather news articles belong to the same event phase into clusters. A relevance optimization algorithm is designed to select top-$k$ news articles.
- We conduct extensive experiments and case studies on multiple newswire datasets and events. The results prove the effectiveness of our approach.

The paper structure is as follows. Section 2 gives an overview of related work in two aspects: multi-document summarization and timeline generation. We define the EPONS task formally in Section 3. Sections 4 and 5 describe details of the clustering and ranking algorithms. We present the datasets and the experiments in Section 6. Finally, we conclude our paper and discuss the future work in Section 7.

## 2 Related work

Given a set of news articles with regard to the same event, a variety of approaches have been proposed to generate a simple and clear summary of the event. Most approaches fall into two categories: Multi-Document Summarization (MDS) and Timeline Generation (TG). In this section, we summarize research work in these fields.

### 2.1 Multi-document summarization

The goal of MDS is to extract the most important elements from a collection of documents and reorganize it into a concise and informative sentence collection. Paradigms used to address this problem can be divided into two categories: based on either extraction or abstraction.

Extraction-based methods evaluate sentences on the basis of importance, and those with the highest scores will be extracted. The evaluation process usually requires manually-defined empirical rules of lexical, syntactic and semantic correlations between grams in sentences [17]. In machine learning, extraction-based summarization is usually modeled as a tagging or ranking problem. Conroy and O'Leary [11] address the problem of determining whether a sentence should be included in the summary via Hidden Markov Models. Wan et al. [42] propose a context-free measure called certainty to rank sentences in summarization. Hong and Nenkova [21] design document-independent features to calculate word importance in a document collection. Ren et al. [35] propose a sentence regression framework to selection top-$k$ sentences by modeling the importance and redundancy simultaneously. Yan et al. [48] detect and summarize multiple events in a news article collection by combinational optimization. In addition, graph-based methods are proved to be effective for ranking tasks, such as cluster-based link analysis [41], LexRank [16], topic graphs [31], etc. Improved topic models can increase performance of event detection and tracking in news streams. For example, Peng et al. [33] propose a Central Topic Model (CenTM) to track dynamic topics in microblog streams. The difference between their work and ours is that we consider mining more fine-grained event phases in a long, complicated event. Additionally, we focus on summarizing event phases from news headlines while CenTM pays more attention to topic trend prediction.

More recently, deep learning based models have verified the effectiveness of learning the representation of sentences in the embedding space. To name a few, He et al. [20] model each sentence in document collection as a linear combination of summary via a sparse coding approach. PriorSum [5] is a convolutional neural network based system to learn summary prior, which is the degree that how much a sentence is appropriate to be selected into summary.

Abstraction-based methods employ the technique of natural language generation to create comprehensive summaries expressed in a more natural way. Qian and Liu [34] utilize fine-grained elements such as words and phrases to generate more specific summaries. Li et al. [27] use event information extraction and abstract representation to generate multi-document summarization for event-oriented news texts. Chopra et al. [10] propose a conditional recurrent neural network to generate shorter versions of sentences based on a convolutional attention-based encoder.

Although MDS is effective for event summarization, when it comes to long, complicated events with large news datasets, these free-text summaries are still difficult for humans to understand within seconds. Nevertheless, MDS has close connections with our work. The graph based representation of news collection in this paper is inspired by graph based summarization [16, 31, 41]. The features used in this paper are also similar to [11, 21], etc.

## 2.2 Timeline generation

Another line of research called TG targets at generating component summaries ordered by time, which is especially suitable for evolutionary news. Timelines can be generated if we simply apply MDS on smaller collection of news articles grouped by publication dates. However, there exist temporal constraints among these component summaries to characterize the development of the events in these news articles. These constraints are not modeled in the above MDS approaches [47].

To generate timelines, Yan et al. [47] utilize trans-temporal characteristics of component summaries and generate news evolution along the timeline by temporal projection. Li and Li [26] propose an evolutionary Hierarchical Dirichlet Process as a generative model to produce timelines. Bauer and Teufel [2] create timelines for Wikipedia history articles based on lexical cues and time expressions. For social network based TG, Zhao et al. [50] consider the social attention from Twitter that reflects users' collective interests, which is integrated into a unified framework for TG.

The dates and headlines of news articles provide additional information for timelines, which is fundamentally different from MDS. Date selection is a subtask of TG which selects the most important dates for an event within a certain time period. Kessler et al. [23] recognize and normalize temporal expressions and then extract salient dates that are relevant to a specific topic. Tran et al. [39] propose supervised and unsupervised joint graphical models for date selection. Based on the fact that headlines of news articles are brief outlines for the contents, Tran et al. [38] exploit influence-based random walk to generate timelines directly from headlines. This work inspires us to produce event phase summaries based on news headlines. Besides, MDS and TG are not completely independent tasks because share similar techniques. For example, Ng et al. [30] integrate temporal information in the form of timelines for the MDS task. Experiments show that the participation of timelines can enhance the performance of MDS.

In summary, both MDS and TG help to generate concise and informative summaries for readers. But for complicated events with long period, it is important to break down the events into different phases to show the evolution process. The identification and summarization of event phases can provide a research foundation for deeper analysis and better understanding of complicated events in the future.

## 3 Problem formulation

A news article $d_i$ can be modeled as a triple $d_i = (h_i, t_i, a_i)$ where $h_i$, $t_i$ and $a_i$ denote the news headline, the publication time/date, and the sentence collection of the content of the article, respectively. A news collection is represented as a set of news articles $D = \{d_i\}$.

In classical aging theory, the life cycle of an event is modeled as four stages: birth, growth, decay and death [7]. However, in a real-life, complicated event, it is difficult to capture the characteristics of the event using only four stages. For example, Figure 1 illustrates the numbers of articles per month per event in our datasets. As can be seen, the development trends and the "peaks" vary significantly across different events. Thus we suggest that a uniform model such as [7] is not suitable for massive news data reporting hot events and topics.

To address this issue, we treat an event as a collection of multiple event phases. Because headlines in news articles are more informative and concrete than the contents [38], in this paper, we use news headlines and publication times to form timelines. We define the concept of an event phase timeline as follows:

**Definition 1 Event Phase Timeline.** An event phase timeline $P$ is a collection of $k$ news headline and publication time pairs ordered by publication time, denoted as $P = \{(h_i, t_i)\}_{i=1}^{k}$ where $t_1 < t_2 < \cdots t_k$.

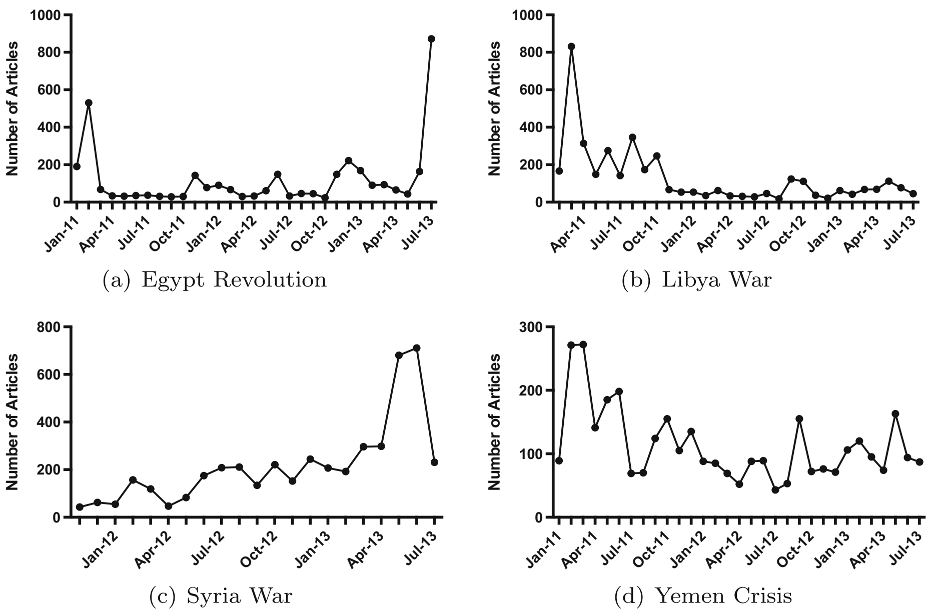



**Figure 1** Number of articles per month per event in our datasets

Event phases are unknown before the event phase timeline generation process, thus need to be identified beforehand. The task of EPONS is defined as follows:

**Definition 2 Event Phase Oriented News Summarization.** Given a news collection $D$ and a positive integer $k$, the goal of EPONS is to generate a collection of $N$ event phase timelines $\mathbf{P} = \{P_j\}_{j=1}^{N}$ where $P_j$ is an event phase timeline w.r.t. the $j$th event phase, i.e., $P_j = \{(h_i, t_i)\}_{i=1}^{k}$.

Based on the definition, the number of event phases $N$ is not pre-defined for an event. Therefore, given a news collection regarding any event, we can produce multiple timelines as a more fine-grained event representation.[2]

Important notations used in this paper are summarized in Table 1.

## 4 Event phase oriented news clustering

In this section, we present our approach for event phase oriented news clustering in detail. The high-level framework is illustrated in Figure 2.

The major challenge is to determine how to measure the degree that two news articles report the same event phase so that they can be grouped into the same cluster. Here, we consider two key factors in terms of content space and time by defining two semantic relations between news articles. Next, the collection of news articles is mapped into a graph representation TCCG which captures the local semantic relations among these articles. A structural clustering algorithm EPCluster separates news articles into candidate event phases by partitioning TCCG into several subgraphs after noise removal. To achieve higher accuracy, an additional postprocessing step is used to filter out clusters that are not related to event phases via a logistic regression classifier. In the following, we will present details of the proposed approach.

### 4.1 Semantic relations between news articles

Relations have been extensively employed to model the semantic connections between entities [22, 37]. However, little has been done to define relations between news articles. In this subsection, we study the characteristics of news articles, and introduce two relations, namely content coherence and temporal influence.

#### *4.1.1 Content coherence*

If two news articles are related to the same event phase, they are not necessarily similar in contents due to difference in reported aspects and writing styles. Different from traditional measures such as VSM with TF-IDF weights (which suffers from curse of dimensionality) or sentence similarity measures [18], we define the content coherence relation considering both topic level and entity level similarity. We calculate the strength of the relation by a content coherence score, denoted as $w_c(d_i, d_j) \in [0, 1]$.

[2] One issue that needs to be discussed here is that because our dataset is relatively large and there are over $k$ news articles in each cluster regarding an event phase, we set a uniform parameter $k$ for all the event phases. We can also modify the definition such that $k$ varies for different event phases without changing our algorithm.

**Table 1** Important notations

| Notation | Description |
|---|---|
| $d_i = (h_i, t_i, a_i)$ | A news article with headline $h_i$, publication time $t_i$ and contents $a_i$ |
| $D = \{d_i\}$ | A news article collection |
| $P = \{(h_i, t_i)\}_{i=1}^{k}$ | An event phase timeline with $k$ components |
| $w_c(d_i, d_j)$ | Content coherence score between $d_i$ and $d_j$ |
| $w_t(d_i, d_j)$ | Temporal influence score between $d_i$ and $d_j$ |
| $G_D = (V, E)$ | *Temporal Content Coherence Graph* w.r.t. news article collection $D$ |
| $C = \{d_i\}$ | A cluster of news articles related to the same event phase |
| $\mathbf{C} = \{C_i\}$ | A cluster collection, which is a partition of $D$ |
| $F(C_i)$ | Quality metric vector of cluster $C_i$ |
| $\mathbf{C}^*$ | A subset of $\mathbf{C}$, corresponding to real event phases |
| $G_{C_i} = (V_{C_i}, E_{C_i})$ | A subgraph of $G_D$ w.r.t. cluster $C_i$ |
| $\mathbf{R_n}$ | Rank vector of news articles in the $n$th iteration |
| $r(d_i)$ | Final rank of news article $d_i$ |
| $r_0(d_i)$ | Prior rank of news article $d_i$ |
| $S_i$ | Selected new article collection w.r.t. the $i$th event phase |

Based on the previous research, it is found that in a stream of news articles, there is a change in distribution of topics over time called topic drift [25]. We regard it as a signal for identifying the change in event phases. To learn the topics, we employ Latent Dirichlet

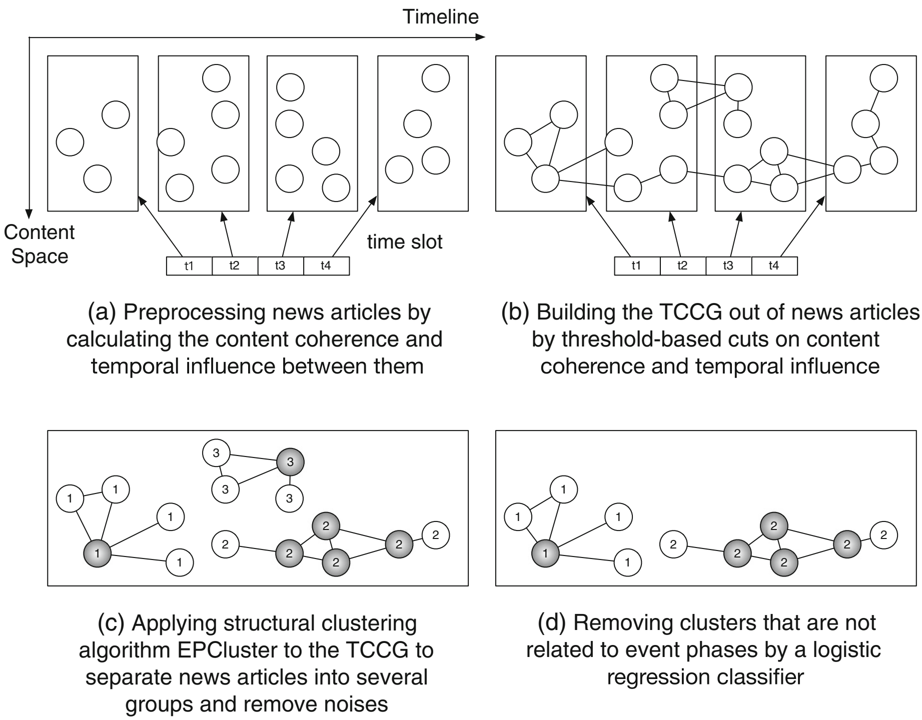



**Figure 2** General framework of event phase oriented news clustering

Allocation (LDA), a well-established topic model for documents [3]. For each news article $d_i \in D$, LDA associates it with a topic distribution vector $\theta_i$. For two articles $d_i$ and $d_j$, the difference between topic distributions are captured by Jansen-Shannon divergence, defined as:

$$D_{JS}(\theta_i \| \theta_j) = \frac{D_{KL}(\theta_i \| \overline{\theta}) + D_{KL}(\theta_j \| \overline{\theta})}{2}$$

where $\overline{\theta} = \frac{\theta_i + \theta_j}{2}$ is the average topic distribution of $d_i$ and $d_j$, and $D_{KL}(\theta_i \| \theta_j)$ is the KL divergence between $\theta_i$ and $\theta_j$. We set $n = 2$ in the base of logarithm for KL divergence to ensure $D_{JS}(\theta_i \| \theta_j) \in [0, 1]$.

Another observation is that, named entities (e.g. people, locations and organizations) play a vital role in news reports, especially for hot topic detection and analysis [40, 51]. If an event goes through different phases, the statistics about these entities are likely to change. Due to the unstructured nature of texts, noisy, incorrect or unnormalized entities will be extracted if we directly apply NER (Named Entity Recognition) techniques. Instead, we utilize our NERank algorithm [43, 45] to extract key entities in the news collection $D$, denoted as $E_D$. Let $\eta_i$ be an $|E_D|$-dimensional count vector of entity collection $E_D$ in news article $d_i$. The entity level similarity between $d_i$ and $d_j$ is calculated by Tanimoto coefficient:

$$TC(\eta_i, \eta_j) = \frac{\eta_i^T \cdot \eta_j}{||\eta_i||^2 + ||\eta_j||^2 - \eta_i^T \cdot \eta_j}$$

Therefore, the content coherence score between $d_i$ and $d_j$ is defined as follows:

$$w_c(d_i, d_j) = \alpha \cdot (1 - D_{JS}(\theta_i \| \theta_j)) + \beta \cdot TC(\eta_i, \eta_j)$$

where $\alpha$ and $\beta$ are tuning parameters that control the strength of entity level and topic level similarity measures. We require $\alpha, \beta \in [0, 1]$ and $\alpha + \beta = 1$.

#### *4.1.2 Temporal influence*

Content coherence alone is not sufficient because it does not capture the temporal dynamics of news. Consider the previous example of Egypt Revolution. There were news articles published in 2011 and 2012 regarding the street protests in Tahrir Square, Cairo. However, although similar in topics and entities, they were in fact related to different event phases, i.e., protests against Hosni Mubarak and the military government, respectively.

The temporal influence relation models the phenomenon that if publication time of $d_i$ and $d_j$ is close, they are likely to report the same event phase. Here, we define the temporal influence score $w_t(d_i, d_j)$ to reflect the strength of the relation by mapping the publication time gap between $d_i$ and $d_j$ into a different space using kernel density estimation. Given $d_i$ and $d_j$, the publication time gap is calculated by $\Delta t_{i,j} = |t_i - t_j|$. We employ the Hamming (cosine) kernel $\Gamma(\cdot)$ [13] to map $\Delta t_{i,j}$ to a real number in [0, 1]:

$$\Gamma(\Delta t_{i,j}) = \begin{cases} \frac{1}{2}(1 + \cos(\frac{\Delta t_{i,j} \cdot \pi}{\sigma})) & (\Delta t_{i,j} \leq \sigma) \\ 0 & (\Delta t_{i,j} > \sigma) \end{cases}$$

where $\sigma$ is a parameter that controls the spread of kernel curves. If $\Delta t_{i,j} > \sigma$, it assumes that there is no direct temporal influence between $d_i$ and $d_j$. Therefore, the temporal influence score is $w_t(d_i, d_j) = \Gamma(\Delta t_{i,j})$.[3]

[3] In the implementation, we set one day as a time slot and compute $w_t(\cdot)$ based on publication date difference. See Figure 2a and b.

### 4.2 EPCluster: a structural news clustering algorithm

With the semantic relations between two news articles properly defined, we now present the EPCluster algorithm in detail, which is a structural algorithm based on TCCG.

#### 4.2.1 Temporal Content Coherence Graph

A first issue to be considered is that given two relation strength scores $w_c(d_i, d_j)$ and $w_t(d_i, d_j)$, how can we determine there is a strong semantic relation between $d_i$ and $d_j$? In this paper, we introduce two parameters $\mu_1$ and $\mu_2$ where $\mu_1, \mu_2 \in (0, 1)$. We say $d_i$ and $d_j$ are directly semantic related iff $w_c(d_i, d_j) > \mu_1$ and $w_t(d_i, d_j) > \mu_2$. In this way, news articles in $D$ can be interconnected and form an undirected graph. See the example in Figure 2b. Here, we define TCCG as follows:

**Definition 3 Temporal Content Coherence Graph.** A Temporal Content Coherence Graph w.r.t. parameters $\mu_1$ and $\mu_2$ and news collection $D$ is an undirected graph $G_D = (V, E)$ such that:

– $V$ is the set of nodes where each node $v_i \in V$ represents a news article $d_i \in D$;
– $E$ is the set of undirected edges where $(v_i, v_j) \in E$ iff $w_c(d_i, d_j) > \mu_1$ and $w_t(d_i, d_j) > \mu_2$.[4]

#### 4.2.2 EPCluster algorithm

Structural clustering has been extensively exploited to summarize and analyze various types of networks [6, 36, 46]. Based on the definition of TCCG, we can extend structural clustering techniques for news clustering. The high-level procedure of EPCluster is illustrated in Algorithm 1.

EPCluster takes three parameters as input, namely $\mu_1$, $\mu_2$ and $MinPts$, where $\mu_1$ and $\mu_2$ are similarity thresholds, which are employed to construct the TCCG given the news article collection $D$. $MinPts$ is the minimum number of objects within $\mu_1$ and $\mu_2$ similarity of an object. Here, we first define the concept of $(\mu_1, \mu_2)$-neighborhood:

**Definition 4 $(\mu_1, \mu_2)$-Neighborhood.** The $(\mu_1, \mu_2)$-neighborhood w.r.t. $d_i$ is a node collection $N(d_i) = \{d_j | (d_i, d_j) \in E\}$.

We can see that $d_j \in N(d_i)$ is equivalent to $w_c(d_i, d_j) > \mu_1$ and $w_t(d_i, d_j) > \mu_2$. In EPCluster, the algorithm categorizes news articles into three types: core, border and noise objects based on $(\mu_1, \mu_2)$-neighborhood, defined as follows:

**Definition 5 Core Object.** A core object is a news article $d_i \in D$ that satisfies $|N(d_i)| \geq MinPts$.

**Definition 6 Border Object.** A border object is a news article $d_i \in D$ that is not a core point and satisfies $d_i \in N(d_j)$ where $d_j \in D$ is a core object.

[4] Based on the definition, we can see that each news article $d_i$ and node $v_i$ has a one-to-one correspondence relationship. In the following, without ambiguity, we will use $d_i$ to represent a node and a news article interchangeably.

**Definition 7 Noise Object.** A noise object is a news article $d_i \in D$ that is neither a core object nor a border object.

The algorithm starts with an object $d_i \in D$ and retrieves all the neighbors in $N(d_i)$ (Line 4). If $d_i$ is a core object, a cluster $C$ (i.e., a news article subset) is created. After that, the cluster is expanded by adding the objects in $d_i$'s neighborhood to the cluster $C$. For each $d_j \in N(d_i)$, if it is a core object, the cluster should be expanded by adding $d_j$'s neighbors to the cluster (Line 6); otherwise, it is a border object. This process continues until a complete cluster $C$ is formed. Thus the algorithm repeats to search for new clusters until all of the objects have been processed. Objects that are not in any cluster are treated as noise objects and discarded.

### *4.2.3 Complexity analysis*

In EPCluster, there is a neighborhood query for each $v_i \in V$, of which the complexity is linearly proportional to $deg(v_i)$ (the degree of $v_i$) with an adjacent list implementation. The entire runtime complexity is $O(\sum_{v_i \in V} deg(v_i))$, which is equivalent to $O(|E|)$. Therefore, the complexity of EPCluster is linear in terms of edges in TCCG.

**Algorithm 1** EPCluster Algorithm

**Input:** News collection $D$, parameters $\mu_1, \mu_2, MinPts$.
**Output:** Cluster collection $\mathbf{C}$.
1: $\mathbf{C} = \emptyset$, $clusterID = 1$;
2: **for** each $d_i \in D$ **do**
3:     **if** $d_i$ is not visited **then**
4:         $N(d_i)$ =SearchNeighbors($d_i, \mu_1, \mu_2$);
5:         **if** $|N(d_i)| \geq MinPts$ **then**
6:             $C_{clusterID}$=ExpandCluster($d_i, \mu_1, \mu_2, MinPts$);
7:             $\mathbf{C} = \mathbf{C} \cup \{C_{clusterID}\}$;
8:             $clusterID = clusterID + 1$;
9:         **end if**
10:     **end if**
11: **end for**
12: **return** $\mathbf{C}$;

## 4.3 Cluster postprocessing

We notice that a few clusters generated by EPCluster do not necessarily represent event phases. Instead, they are "small" clusters with similar articles. To improve the accuracy of event phase extraction, we design a quality assessment function to filter such clusters. We consider the following four quality metrics:

**Percentage of New Articles** The first quality metric measures the size of the cluster by the percentage of news articles in that cluster. For cluster $C_i \in \mathbf{C}$, the metric is calculated as: $N(C_i) = \frac{|C_i|}{|D|} \times 100\%$.

**Time Interval** For cluster $C_i \in \mathbf{C}$, denote $(t_1^i, t_2^i, \cdots, t_{|C_i|}^i)$ as the sequence of publication dates sorted chronologically. Let $t_{Q1}^i$ and $t_{Q3}^i$ be the first and third quantiles of the empirical temporal distribution. Based on the statistics theory, we estimate the time interval of $C_i$ as $T(C_i) = t^i_{max} - t^i_0$ where

$$t^i_0 = \max\{t^i_1, t^i_{Q1} - 1.5 \cdot |t^i_{Q3} - t^i_{Q1}|\}$$

$$t^i_{max} = \min\{t^i_{|C_i|}, t^i_{Q3} + 1.5 \cdot |t^i_{Q3} - t^i_{Q1}|\}$$

**Pairwise Topic Similarity** Articles reporting the same phase should be similar in topic distributions. We define the average pairwise topic similarity as a quality metric:

$$ATS(C_i) = 1 - \frac{2\sum_{d_m,d_n \in C_i (m<n)} D_{JS}(\theta_m \| \theta_n)}{|C_i| \cdot (|C_i| - 1)}$$

**Pairwise Entity Similarity** Similarly, we define the average pairwise entity similarity as follows:

$$AES(C_i) = \frac{2\sum_{d_m,d_n \in C_i (m<n)} TC(\eta_m, \eta_n)}{|C_i| \cdot (|C_i| - 1)}$$

For each cluster $C_i$, we generate a feature vector consisting of four quality metrics: $F(C_i) =< N(C_i), T(C_i), ATS(C_i), AES(C_i) >$. A weight vector $\mathbf{s}$ gives different weights for each feature in $F(C_i)$. Therefore, for each cluster $C_i$, we define a score function $Score(C_i) = \mathbf{s} \cdot F(C_i)$ to indicate the degree that it is related to an event phase. To classify the clusters based on the score function, we construct a logistic regression classifier, with the prediction function as follows:

$$f(C_i) = \frac{1}{1 + e^{-\mathbf{s} \cdot F(C_i)}}$$

We use logistic regression as the classifier because it is simple thas relatively high performance in practice. Because the amount of data in the postprocessing step (i.e., statistics of clusters) is not large. Using such classifiers can learn the parameters well and avoids overfitting. Hence we learn the weight vector $\mathbf{s}$ via gradient ascent on a labeled dataset. After the model $f$ is trained, we can filter out a news cluster $C_i$ if $f(C_i) < 0.5$. The rest of the clusters (denoted as $\mathbf{C}^*$) are corresponding to event phases.

## 5 Event phase summarization

In this section, we introduce our steps to generate event phase summaries based on the previous clustering results. While the relevance between a news article and an event (represented as keywords e.g. Egypt Revolution) can be easily estimated by IR techniques, it is challenging to determine which articles are more relevant to an event phase. In this paper, we design a vertex-reinforced random walk based approach to calculate the relevance scores. Event phase summaries can be generated by relevance maximum optimization with constraints.

### 5.1 News article ranking

For each $C_i \in \mathbf{C}^*$, we construct a subgraph $G_{C_i} = (V_{C_i}, E_{C_i})$ out of the TCCG $G_D$. $d_j \in V_{C_i}$ iff $d_j \in C_i$. $(d_j, d_k) \in E_{C_i}$ iff $d_j \in C_i$, $d_k \in C_i$ and $(d_j, d_k) \in E$. Refer to a simple example in Figure 2d.

While the standard PageRank algorithm [4] employs a time-homogeneous random walk process on a graph, it tends to assign high scores to closely connected communities, which is capable of selecting nodes with high centrality [12]. To generate representative articles that better summarize the event phase, we need to pay attention to diversity as well. We adopt the vertex-reinforced random walk process framework [29, 32] to balance centrality and diversity in ranking.

In vertex-reinforced random walk process, denote $M_{m,n}^{(0)}$ as the prior transition probability from $d_m$ to $d_n$. $N_p(n)$ is the number of visits of random walker up to the $p$th iteration. The transition probability from $d_m$ to $d_n$ in the $(p+1)$th iteration is $M_{m,n}^{(p+1)} \propto M_{m,n}^{(p)} N_p(n)$. Therefore, $M_{m,n}^{(p+1)}$ is reinforced by $N_p(n)$. This results in a "rich-gets-richer" effect on ranking scores in a community.

In this paper, we calculate the relevance scores of news articles by extending the vertex-reinforced random walk to the subgraph of TCCG. The implementation is shown in Algorithm 2. Denote $\mathbf{R_0}$ as a $|C_i| \times 1$ prior ranking vector for articles in $C_i$. In our previous work [44], $\mathbf{R_0}$ is set uniformly, i.e., $\mathbf{R_0} = \frac{1}{|C_i|}\mathbf{e}$ where $\mathbf{e}$ is a $|C_i| \times 1$ vector with all elements assigned to 1. We additionally introduce a headline ranking model to compute $\mathbf{R_0}$, which will be detailed in Section 5.2. Denote $\mathbf{R}^*$ as the rank vector calculated based on the headline ranking model. For transition probabilities, $M_{m,n}^{(0)}$ (the element in the $m$-th row and $n$-th column of the prior transition matrix $\mathbf{M_0}$) is defined using the fusion of the two relation strength scores:

$$M_{m,n}^{(0)} = \begin{cases} \frac{1}{Z} \cdot w_c(d_m, d_n) \cdot w_t(d_m, d_n) & (d_m, d_n) \in E_{C_i} \\ 0 & otherwise \end{cases}$$

where $Z$ is a normalization factor and $\lambda$ is a damping factor, typically set to 0.85. Let $\mathbf{M_{p+1}}$ be the transition probability matrix in the $(p+1)$th iteration, which is updated according to the ranking values and transition probability matrix in the previous iteration:

$$\mathbf{M_{p+1}} = \lambda \mathbf{T_p} \cdot \mathbf{M_p} + (1-\lambda)\mathbf{M_0}$$

where $\mathbf{T_p} = [\mathbf{R_p} \mathbf{R_p} \cdots \mathbf{R_p}]$ is a $|C_i| \times |C_i|$ matrix which is utilized to update the transition matrix based on the ranking values in the previous iteration. The update rule for ranking values is defined as:

$$\mathbf{R_{p+1}} = \lambda \mathbf{M_{p+1}} \cdot \mathbf{R_p} + (1-\lambda)\mathbf{R_0}$$

The above iterative formula defines an ergodic random walk process in a Markov chain. As shown in [29], it also converges to a stationary distribution. After sufficient large times of iteration $N^*$, we obtain

$$r(d_m) = \sum_{d_n \in C_i} M_{m,n}^{(p)} \cdot r(d_n)$$

as the relevance score of $d_j$ when $p > N^*$.

**Algorithm 2** News Article Ranking Algorithm

**Input:** News cluster $C_i$, parameter $\lambda$.
**Output:** Ranking vector $\mathbf{R}$.
1: Compute the prior transition matrix $\mathbf{M_0}$ based on $C_i$;
2: Compute the prior rank matrix $\mathbf{R_0} = \mathbf{R}^*$ based on the headline ranking model;
3: Initialize the iteration counter $p = 0$;
4: **while** not converge **do**
5: $\quad \mathbf{T_p} = [\mathbf{R_p R_p} \cdots \mathbf{R_p}]$;
6: $\quad \mathbf{M_{p+1}} = \lambda \mathbf{T_p} \cdot \mathbf{M_p} + (1-\lambda)\mathbf{M_0}$;
7: $\quad \mathbf{R_{p+1}} = \lambda \mathbf{M_{p+1}} \cdot \mathbf{R_p} + (1-\lambda)\mathbf{R_0}$;
8: $\quad p = p + 1$;
9: **end while**
10: **return** $\mathbf{R} = \mathbf{R_p}$;

### 5.2 Headline ranking model

The headline ranking model gives prior knowledge about whether a headline is suitable to be an entry in a timeline. In a vast of news articles, some news headlines are informing and describe real-life events that have occurred. Others express the opinions or subjective views of the journalists on the event, which are not appropriate to be appeared in timelines.

Inspired by the previous work [38], we employ a model based approach to assign a prior rank $r_0(d_i)$ to news article $d_i$. To predict whether a headline reports about facts or opinions, we train a Naive Bayes classifier based on word features, which has high performance in [38, 49]. For a news article $d_j$ from the cluster $C_i$, denote $p(d_j)$ as the probability of $d_j$ being a factual headline. We calculate the prior rank of $d_j$ as follows:

$$r_0(d_j) = \frac{p(d_j) + \gamma}{\sum_{d_{j'} \in C_i} (p(d_{j'}) + \gamma)}$$

where $\gamma$ serves as a smoothing factor.

### 5.3 Event phase summary generation

The final step of our method is to generate an event summary $P_i$ by extracting headlines and publication time of $k$ selected news articles (denoted as $S_i$). We formulate the news article selection task as an optimization problem that can be solved by a greedy algorithm.

Ideally, the selected news articles must be relevant to the event phase. However, we notice that the generated summary must not contain redundant information. Therefore, we add an additional constraint such that for any two selected news articles $d_m$ and $d_n$, we require $w_c(d_m, d_n) \leq \mu_1$ and $w_t(d_m, d_n) \leq \mu_2$. Our News Selection optimization problem is defined as the following optimization problem:

$$\begin{aligned}
&\max_{S_i \subset C_i} R(S_i) = \sum_{d_j \in S_i} r(d_j) \\
&\text{subject to} |S_i| = k \\
&\qquad \forall d_m, d_n \in S_i, w_c(d_m, d_n) \leq \mu_1 \\
&\qquad \forall d_m, d_n \in S_i, w_t(d_m, d_n) \leq \mu_2
\end{aligned}$$

The proposed optimization problem can be seen as a special case of the budgeted maximum coverage problem [24], which is proved to be NP-hard. Because the optimization objective is submodular and monotone, we can employ a greedy algorithm to solve the problem approximately. Here, we present our approximate algorithm for News Selection in Algorithm 3. The worst-case approximation ratio is proved to be $1 - \frac{1}{e}$, as shown by Khuller et al. [24]. It selects a news article from $S_i$ that maximizes that objective function without violating any constraints at each iteration. When it stops with $k$ news articles selected, we extract the publication time and headlines in $S_i$ as the event phase summary $P_i$.

**Algorithm 3** News Article Selection Algorithm

**Input:** News cluster $C_i$, parameter $k$.
**Output:** Selected news collection $S_i$.

$S_i = \emptyset$;
**while** $C_i \neq \emptyset$ and $|S_i| < k$ **do**
    Select $d_n = \text{argmax}_{d_n \in C_i} R(S_i \cup \{d_n\}) - R(S_i)$
    subject to $w_c(d_m, d_n) \leq \mu_1,\ w_t(d_m, d_n) \leq \mu_2, \forall d_m \in S_i$;
    $S_i = S_i \cup \{d_n\}$;
    $C_i = C_i \setminus \{d_n\}$;
**end while**
**return** $S_i$;

## 6 Experimental results

In this section, we conduct experiments on news datasets to evaluate the performance of our approaches under two frameworks: news clustering and timeline generation. We compare our approaches with baselines to make the convincing conclusion.

### 6.1 Datasets

The news datasets we used in this paper are publicly available from [38]. They contain four English news datasets regarding long-span recent armed conflicts. The news articles are collected from 24 news agencies (e.g. Associated Press, Reuters, Guardian, etc.), obtained using Google search engine. The detailed statistics are illustrated in Table 2.

### 6.2 Evaluation on event phase oriented news clustering

We first report the performance of our news clustering method *EPCluster* and its postprocessing step.

**Table 2** Summary of datasets

| Dataset | Event | Number of articles | Time range |
|---|---|---|---|
| $D_1$ | Egypt revolution | 3,869 | 2011.1.11 – 2013.7.24 |
| $D_2$ | Libya war | 3,994 | 2011.2.16 – 2013.7.18 |
| $D_3$ | Syria war | 4,071 | 2011.11.17 – 2013.7.26 |
| $D_4$ | Yemen crisis | 3,600 | 2011.1.15 – 2013.7.25 |

### 6.2.1 Experimental settings

To our knowledge, there is no prior work regarding event phase oriented news clustering. However, the proposed approach can be seen as an application of document clustering. To obtain the ground truth, we employ a pairwise judgment method introduced in [8]. For each dataset $D_i$, we randomly generate news article pairs, denoted as $L_i = \{(d_m, d_n)\}$. We ask human annotators to label whether $d_m$ and $d_n$ are related to the same event phase. Denote $v_{m,n} \in \{1, 0\}$ as the human judgment result and $v'_{m,n}$ as the clustering result, where 1 and 0 represent the same and different phases, respectively. We use precision, recall and F1 score as the evaluation metrics, defined as:

$$
\begin{aligned}
Precision(L_i) &= \frac{|\{(d_m, d_n) \in L_i | v_{m,n} = 1 \wedge v'_{m,n} = 1\}|}{|\{(d_m, d_n) \in L_i | v'_{m,n} = 1\}|} \\
Recall(L_i) &= \frac{|\{(d_m, d_n) \in L_i | v_{m,n} = 1 \wedge v'_{m,n} = 1\}|}{|\{(d_m, d_n) \in L_i | v_{m,n} = 1\}|} \\
F1\ Score(L_i) &= \frac{2 \cdot Precision(L_i) \cdot Recall(L_i)}{Precision(L_i) + Recall(L_i)}
\end{aligned}
$$

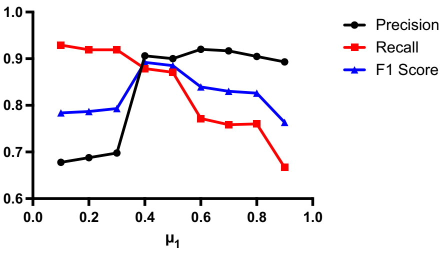
In total, we have 300 labeled new article pairs for each dataset. We use 30% of the labeled data for parameter tuning and the rest for testing. We report the macro-average precision, recall and F1 score in the following experiments.

### 6.2.2 Parameter analysis

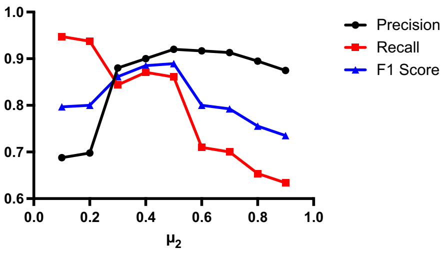
We tune three parameters in EPCluster, namely $\mu_1$, $\mu_2$ and $MinPts$. For simplicity, we set $\alpha = \beta = \frac{1}{2}$ to calculate the content coherence and leave automatic learning the values of $\alpha$ and $\beta$ for future research. In the EPCluster algorithm, we fix two parameters in $\mu_1$, $\mu_2$ and $MinPts$ and vary the remaining one at each time. Due to space limitation, we only report the performance over the development set w.r.t. Egypt Revolution and illustrate the overall performance over all the datasets in the next subsection.

More specifically, we fix $\mu_1 = 0.5$, $MinPts = 10$ and vary $\mu_2$ from 0.1 to 0.9 in Figure 3a. We fix $\mu_2 = 0.5$, $MinPts = 10$ and vary $\mu_1$ from 0.1 to 0.9 in Figure 3b. In Figure 3c, we fix $\mu_1 = \mu_2 = 0.5$ and vary $MinPts$ from 5 to 30. From the experimental results, we can see that $\mu_1$ and $\mu_2$ have similar impacts on the effectiveness of EPCluster. When $\mu_1$ and $\mu_2$ are too small, EPCluster will produce large numbers of small clusters. News articles that are related to the same event phase are separated into different clusters. On the contrary, when $\mu_1$ and $\mu_2$ are too large, small numbers of large clusters will be produced, forcing news articles related to different event phases to be merged together. The parameter $MinPts$ also controls the number of clusters by constraining the size of the $(\mu_1, \mu_2)$-neighborhood. Overall, it can be seen that when $\mu_1 = 0.4$, $\mu_2 = 0.5$ and $MinPts = 10$, EPCluster achieves the best results over the development set.

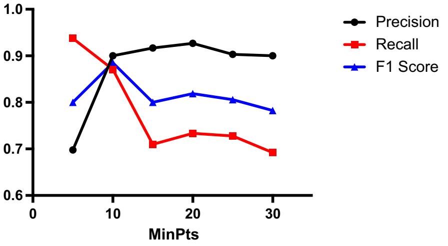
To train the logistic regression model in the postprocessing step, we run the EPCluster multiple times with different parameter settings and label 200 clusters in the development phase. The F-measure is 97.3%, indicating the effectiveness of the model. We will report how this method can reduce the number of "meaningless" clusters in the next subsection.




(a) Varying $\mu_1$




(b) Varying $\mu_2$




(c) Varying $MinPts$

**Figure 3** Clustering results of *EPCluster* under different parameter settings

### *6.2.3 Study on the number of event phases*

While it is relatively easy to judge whether two news articles are related to the same event phase, it is difficult to determine the number of event phases given a news collection. In this subsection, we present a preliminary study on this part. For each event, we ask human annotators to check the corresponding pages in Wikipedia. Based on the detailed descriptions of these events, the numbers of event phases are determined by majority voting of human annotators. We treat these numbers as ground truth and leave more rigorous evaluation methods for future research.

In Table 3, we compare the numbers of event phases generated by EPCluster with and without the postprocessing step, and those by ground truth. As seen, in most cases, the cluster numbers exactly equal to the ground truth. For the event of Syria War, our algorithm generates 8 clusters rather than 6 clusters. The most possible cause is that Syria War is a continuous event, with no clear boundaries between these phases. Comparing the numbers of clusters with and without the postprocessing step, we can conclude that this step is effective to reduce small and “meaningless” clusters.

**Table 3** Numbers of event phases generated by our approach and ground truth

| Event | #Clusters (EPCluster) | #Clusters (EPCluster+Post) | Ground truth |
|---|---|---|---|
| Egypt revolution | 10 | 5 | 5 |
| Libya war | 11 | 3 | 3 |
| Syria war | 12 | 8 | 6 |
| Yemen crisis | 10 | 6 | 6 |

### 6.2.4 End-to-end clustering performance evaluation

In this subsection, we report the performance of our method on the test set. We also compare our method with classical document clustering approaches and the variant of our method, introduced as follows:

- **VSMCluster** - It uses the KMeans algorithm [19] for clustering based on word features of TF-IDF weights.
- **TopicCluster** - It uses the KMeans algorithm [19] for clustering based on document topic distributions generated by LDA [3].
- **SCAN** - It use the structural clustering algorithm SCAN [46] for network partitioning in the TCCG.
- **EPCluster** - It is our EPCluster algorithm without postprocessing.
- **EPCluster+Post** - It is our EPCluster algorithm with the postprocessing step.

In the implementation, because we consider publication time in EPCluster, we add it as a feature in VSMCluster and TopicCluster to make them comparable with ours. The results of these methods are presented in Table 4.

**Table 4** Experimental results of event phase oriented news clustering

| Method | Precision | Recall | F1 Score |
|---|---|---|---|
| Event: Egypt revolution | | | |
| VSMCluster | 0.45 | 0.69 | 0.54 |
| TopicCluster | 0.58 | 0.69 | 0.63 |
| SCAN | 0.80 | 0.75 | 0.77 |
| EPCluster | 0.83 | 0.82 | 0.82 |
| EPCluster+Post | 0.90 | 0.81 | 0.85 |
| Event: Libya war | | | |
| VSMCluster | 0.38 | 0.75 | 0.50 |
| TopicCluster | 0.54 | 0.72 | 0.62 |
| SCAN | 0.81 | 0.73 | 0.75 |
| EPCluster | 0.81 | 0.77 | 0.78 |
| EPCluster+Post | 0.91 | 0.77 | 0.83 |
| Event: Syria war | | | |
| VSMCluster | 0.29 | 0.72 | 0.41 |
| TopicCluster | 0.45 | 0.62 | 0.53 |
| SCAN | 0.77 | 0.73 | 0.75 |
| EPCluster | 0.77 | 0.76 | 0.77 |
| EPCluster+Post | 0.85 | 0.77 | 0.80 |
| Event: Yemen crisis | | | |
| VSMCluster | 0.28 | 0.80 | 0.41 |
| TopicCluster | 0.50 | 0.65 | 0.57 |
| SCAN | 0.77 | 0.73 | 0.75 |
| EPCluster | 0.83 | 0.81 | 0.82 |
| EPCluster+Post | 0.90 | 0.78 | 0.83 |

Based on the experimental results, our method outperforms VSMCluster and TopicCluster because these classical methods rely on distance computation of high-dimensional vectors. Since these news articles are related to the same event therefore similar in contents, these methods are not suitable for event phase oriented news clustering. The SCAN algorithm has a relatively good performance based on TCCG, which indicates that although structural clustering is originally designed for networks, it can be employed for text analysis as well. The comparison between EPCluster and EPCluster+Post shows that the postprocessing step is effective to improve the performance of event phase oriented news clustering. Overall, the proposed approach has an F1-score of over 80% for all the events.

### 6.3 Evaluation on event phase summarization

We evaluate the performance of event phase summarization under the framework of timeline generation.

### 6.4 Ground truth acquisition

The ROUGE framework [28] has been extensively used to evaluate the effectiveness of document summarization. However, the summaries we generate are headlines, rather than documents. Tran et al. [38] previously propose a headline summary evaluation framework based on the relevance of machine-generated timelines compared with ground truth timelines. The timeline summaries used in this paper were manually created by professional journalists from Tran et al. These timeline summaries are served as ground truth to be provided to human annotators for the evaluation of our method. The detailed statistics of ground truth summaries can be found in [38].

### 6.5 Method comparison

Although there is no prior work addressing the event phase summarization issue, if we consider the single summary of an event phase, our task can be regarded as a headline summary generation task. We compare our method with the following baselines:[5]

- **Tran et al.** [38] - It selects top-$k$ news headlines as the timeline based on headline relevance.
- **Chieu et al.** [9] - It is a timeline generation approach based sentence popularity. We use it to select top-$k$ news articles and then take the headlines as timelines.
- **Our Method (PageRank)** - It is the variant of our approach which adopts simple PageRank method for relevance calculation.

We also consider the following two benchmark methods:

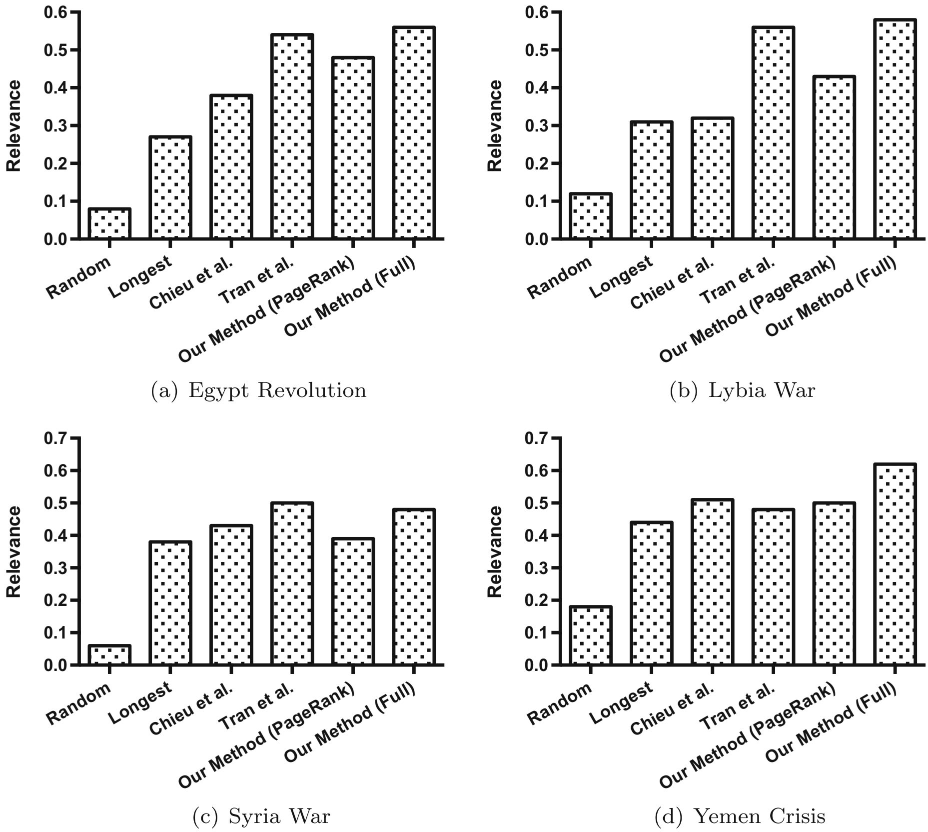
- **Random** - It selects $k$ news headlines randomly.
- **Longest** - It selects top-$k$ longest headlines because longer headlines are more informative compared to others.

[5]Many other methods focus on timeline generation. However, the summaries we generates are headlines and dates, making it difficult to compare our method with them. We will investigate how to modify these algorithms for our task in the future.

### 6.6 Experimental results

To generate timelines as event phase summaries, we annotate a headline dataset sampled from our news datasets which contains 500 news headlines, and train the headline ranking model with $\gamma = 10^{-2}$. The performance is 85.4% in terms of F-measure. Next, we generate timelines based on the proposed news article ranking and selection method. Following [38], we generate summaries from 106 dates that are appeared in the ground truth summaries. For fair comparison, the parameter $k$ is set to the number of entries in the ground truth summary so that the results can be directly compared. We present the ground truth and machine-generated summaries to human annotators and ask them to label each headline as relevant or not. The evaluation method is the same in [38]. We take the average relevance scores for each method as the evolution metrics. The results are presented in Figure 4.

It can be seen that the results of benchmark approaches are not as good as others because they lack textual analysis on news articles. Especially, the method Random has the relevance scores ranging from 5% to 20%, which has the worst performance. Our method outperforms Chieu et al. and the variant of our method because we pay more attention to the centrality and diversity nature of summaries.




(a) Egypt Revolution (b) Lybia War

(c) Syria War (d) Yemen Crisis

**Figure 4** Relevance evaluation of event phase summarization

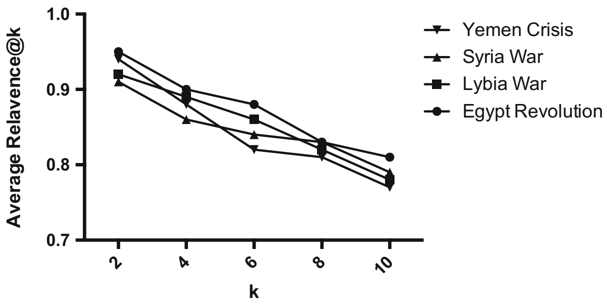



**Figure 5** Parameter analysis of *k*

The performance of Tran et al. is relatively high because they investigate the characteristics of news headlines and select more informative ones. Our method performs slightly better than Tran et al. in terms of relevance for three events. For Syria War, our method is comparable with Tran et al. The unique advantage of ours is that we generate multiple summaries for event phases such that it is easier for readers to track the development phases of long, complicated events.

As a further experiment, we fix the parameter $k$ for each dataset to generate timelines with the same length. We again evaluate the average relevance scores using the same approach. The result is presented in Figure 5. It shows that there is a slow declining trend of average relevance scores when $k$ increases.

**Table 5** Event phase summaries of Egypt Revolution

| | |
|---|---|
| **Event Phase #1** Protest against Hosni Mubarak | |
| 2011.2.2 | Egypt protests: Hosni Mubarak to stand down at next election |
| 2011.2.3 | In Egypt, President Hosni Mubarak still has support, from rich and poor |
| 2011.2.11 | Hosni Mubarak resigns and Egypt celebrates a new dawn |
| **Event Phase #2** Egypt under the Rule of the Military Government | |
| 2011.4.9 | Egyptian soldiers attack Tahrir Square protesters |
| 2011.7.10 | Protests spread in Egypt as discontent with military rule grows |
| 2011.12.18 | Egypt's military clashes with protesters for third day |
| **Event Phase #3** Mohammed Morsi Won Presidential Election | |
| 2012.5.23 | First round of presidential election |
| 2012.6.19 | Islamists claim win, army grabs power in Egypt |
| 2012.6.24 | Election officials declare Morsi the winner |
| **Event Phase #4** Protest against Morsi and Muslim Brotherhood | |
| 2012.12.24 | Egypt opposition vows to overturn Islamist constitution |
| 2013.1.27 | Egypt's Morsi declares state of emergency, curfew after deadly clashes |
| 2013.1.30 | Egypt's military chief says clashes threaten the state |
| **Event Phase #5** Morsi's Ousting | |
| 2013.7.4 | After Morsi's ousting, Egypt swears in new president |
| 2013.7.6 | Morsi's ouster in Egypt sends chill through political Islam |
| 2013.7.6 | Egypt El Baradei to head interim government? |

**Table 6** Event phase summaries of Libya War

| | |
|---|---|
| **Event Phase #1** The First Civil War in Libya | |
| 2011.2.26 | Libya protests 5 shot dead in Tripoli after Gaddafi troops open fire |
| 2011.8.25 | Libya rebels battle to purge Tripoli of Gadhafi loyalists |
| 2011.10.21 | Muammar Gaddafi's body held in old meat locker as debate rages over dictator's burial |
| **Event Phase #2** Attack against US Consulate | |
| 2012.9.12 | Libya ambassador Chris Stevens killed |
| | U.S. warships headed to Libyan coast as Obama says 'justice will be done' |
| 2011.10.16 | Clinton takes responsibility for consulate security, blames confusion on 'fog of war' |
| 2011.10.20 | Gadhafi's youngest son reported killed amid Libya clashes |
| **Event Phase #3** Post-revolution violence and the attack against NTC | |
| 2013.5.8 | Libya crisis deepens as rebel groups expand demands |
| 2013.4.23 | Car bomb explodes at French embassy in Libyan capital |
| 2013.7.1 | Hundreds of British soldiers to deploy to Libya 'within months' to train army in their battle with Al Qaeda extremists |

### 6.7 Case studies

We present the event phase summaries of Egypt Revolution and Libya War produced by our approach. Due to space limitation, we only present the publication dates and headlines of three news articles in each event phase. We also manually add a brief description for each phase, shown in Tables 5 and 6. It shows that our approach can identify and summarize fine-grained event phases effectively. Compared to the corresponding Wikipedia pages, our method gives a broader view of the event. For example, in Table 5, we can see that despite the revolution, President Hosni Mubarak stills had some degrees of support from Egyptians (based on the news headline on February 3rd, 2011). This piece of factual information is not covered in Wikipedia.

## 7 Conclusion and future work

In this paper, we introduce the task of EPONS (Event Phase Oriented News Summarization) to summarize complicated events from the Web. We propose a structural clustering algorithm EPCluster based on TCCG to group news articles into event phases. For each event phase, we extract top-$k$ news articles by a vertex reinforced random walk based ranking algorithm and generate timeline summaries by relevance maximum optimization. Experiments on multiple datasets and events show that our method can cluster news articles and generate timelines effectively, which considers multiple event phases. In the future, we will focus on improving the performance of document summarization when event phases are considered.

**Acknowledgments** This work is partially supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904. Chengyu Wang is partially supported by the Outstanding Doctoral Dissertation Cultivation Plan of Action under Grant No. YB2016040. This manuscript is an extended version of the paper "Event Phase Extraction and Summarization" presented at WISE 2016 [44].

## References


1. Bansal, T., Kanti Das, M., Bhattacharyya, C.: Content driven user profiling for comment-worthy recommendations of news and blog articles. In: Proceedings of the 9th ACM, Conference on Recommender Systems, pp. 195–202 (2015)
2. Bauer, S., Teufel, S.: Unsupervised timeline generation for wikipedia history articles. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2343–2349 (2016)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. **30**(1–7), 107–117 (1998)
5. Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., Wang, H.: Learning summary prior representation for extractive 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, pp. 829–833 (2015)
6. Chang, L., Li, W., Lin, X., Qin, L., Zhang, W.: pscan: Fast and exact structural graph clustering. In: Proceedings of the 32nd IEEE, International Conference on Data Engineering, pp. 253–264 (2016)
7. Chen, C.C., Chen, Y.-T., Sun, Y.S., Chen, M.C.: Life cycle modeling of news events using aging theory. In: Proceedings of the 14th European Conference on Machine Learning, pp. 47–59 (2003)
8. Chen, J., Niu, Z., Fu, H.: A multi-news timeline summarization algorithm based on aging theory. In: Web Technologies and Applications - 17th Asia-Pacific Web Conference, pp. 449–460 (2015)
9. Chieu, H.L., Lee, Y.K.: Query based event extraction along a timeline. In: Proceedings of the 27th Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 425–432 (2004)
10. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Human Language Technologies: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 93–98 (2016)
11. Conroy, J.M., O'Leary, D.P.: Text summarization via hidden markov models. In: Proceedings of the 24th Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 406–407 (2001)
12. Davis, J.V., Dhillon, I.S.: Estimating the global pagerank of Web communities. In: Proceedings of the Twelfth ACM, SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 116–125 (2006)
13. de Kretser, O., Moffat, A.: Effective document presentation with a locality-based similarity heuristic. In: Proceedings of the 22nd Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 113–120 (1999)
14. Diao, Q., Shan, J.: A new Web page summarization method. In: Proceedings of the 29th Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 639–640 (2006)
15. Dolby, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Schonberg, E., Srinivas, K., Ma, L.: Scalable semantic retrieval through summarization and refinement. In: Proceedings of the Twenty-Second AAAI, Conference on Artificial Intelligence, pp. 299–304 (2007)
16. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. (JAIR) **22**, 457–479 (2004)
17. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 19–25 (2001)
18. Gu, Y., Yang, Z., Xu, G., Nakano, M., Toyoda, M., Kitsuregawa, M.: Exploration on efficient similar sentences extraction. World Wide Web **17**(4), 595–626 (2014)
19. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. J. R. Stat. Soc.: Ser. C: Appl. Stat. **28**(1), 100–108 (1979)
20. He, Z., Chen, C., Bu, J., Wang, C., Zhang, L., Cai, D., He, X.: Document summarization based on data reconstruction. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
21. Hong, K., Nenkova, A.: Improving the estimation of word importance for news multi-document summarization. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 712–721 (2014)
22. Jiang, L., Luo, P., Wang, J., Xiong, Y., Lin, B., Wang, M., An, N.: GRIAS: an entity-relation graph based framework for discovering entity aliases. In: Proceedins of the 2013 IEEE, 13th International Conference on Data Mining, pp. 310–319 (2013)

23. Kessler, R., Tannier, X., Hagége, C., Moriceau, V., Bittar, A.: Finding salient years for building thematic timelines. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 730–739 (2012)
24. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. Inf. Process. Lett. **70**(1), 39–45 (1999)
25. Knights, D., Mozer, M.C., Nicolov, N.: Detecting topic drift with compound topic models. In: Proceedings of the Third International Conference on Weblogs and Social Media (2009)
26. Li, J., Li, S.: Evolutionary hierarchical dirichlet process for timeline summarization. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 556–560 (2013)
27. Li, W., He, L., Zhuge, H.: Abstractive news summarization based on event semantic link network. In: Proceedings of the 26th International Conference on Computational Linguistics, pp. 236–246 (2016)
28. Lin, C.-Y., Hovy, E.H.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (2003)
29. Mei, Q., Guo, J., Radev, D.R.: Divrank: the interplay of prestige and diversity in information networks. In: Proceedings of the 16th ACM, SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1009–1018 (2010)
30. Ng, J.-P., Chen, Y., Kan, M.-Y., Li, Z.: Exploiting timelines to enhance multi-document summarization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 923–933 (2014)
31. Parveen, D., Ramsl, H.-M., Strube, M.: Topical coherence for graph-based extractive summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1949–1954 (2015)
32. Pemantle, R.: Vertex-reinforced random walk. Probab. Theory Relat. Fields **92**(1), 117–136 (1992)
33. Peng, M., Zhu, J., Li, X., Huang, J., Wang, H., Zhang, Y.: Central topic model for event-oriented topics mining in microblog stream. In: Proceedings of the 24th ACM, International Conference on Information and Knowledge Management, pp. 1611–1620 (2015)
34. Qian, X., Liu, Y.: Fast joint compression and summarization via graph cuts. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1492–1502 (2013)
35. Ren, P., Wei, F., Chen, Z., Ma, J., Zhou, M.: A redundancy-aware sentence regression framework for extractive summarization. In: Proceedings of the 26th International Conference on Computational Linguistics, pp. 33–43 (2016)
36. Seeland, M., Berger, S.A., Stamatakis, A., Kramer, S.: Parallel structural graph clustering. In: Proceedings of the 2011 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 256–272 (2011)
37. Shen, W., Wang, J., Luo, P., Wang, M.: A hybrid framework for semantic relation extraction over enterprise data. Int. J. Semantic Web Inf. Syst. **11**(3), 1–24 (2015)
38. Tran, G.B., Alrifai, M., Herder, E.: Timeline summarization from relevant headlines. In: Advances in Information Retrieval - 37th European Conference on IR, Research, pp. 245–256 (2015)
39. Tran, G.B., Herder, E., Markert, K.: Joint graphical models for year 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, pp. 1598–1607 (2015)
40. Unankard, S., Li, X., Sharaf, M.A.: Emerging event detection in social networks with location sensitivity. World Wide Web **18**(5), 1393–1417 (2015)
41. Wan, X., Yang, J.: Multi-document summarization using cluster-based link analysis. In: Proceedings of the 31st Annual International ACM, SIGIR Conference on Research and Development in Information Retrieval, pp. 299–306 (2008)
42. Wan, X., Zhang, J.: CTSUM: extracting more certain summaries for news articles. In: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 787–796 (2014)
43. Wang, C., Zhang, R., He, X., Zhou, A.: Nerank: Ranking named entities in document collections. In: Proceedings of the 25th International Conference on World Wide Web, pp. 123–124 (2016)
44. Wang, C., Zhang, R., He, X., Zhou, G., Zhou, A.: Event phase extraction and summarization. In: Proceedings of the 17th International Conference on Web Information Systems Engineering, pp. 473–488 (2016)
45. Wang, C., Zhang, R., He, X., Zhou, G., Zhou, A.: Nerank: Bringing order to named entities from texts. In: Web Technologies and Applications - Proceedings of the 18th Asia-Pacific Web Conference, pp. 15–27 (2016)

46. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: SCAN: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM, SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 824–833 (2007)
47. Yan, R., Kong, L., Huang, C., Wan, X., Li, X., Zhang, Y.: Timeline generation through evolutionary trans-temporal summarization. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 433–443 (2011)
48. Yan, J., Cheng, W., Wang, C., Liu, J., Gao, M., Zhou, A.: Optimizing word set coverage for multi-event summarization. J. Comb Optim. **30**(4), 996–1015 (2015)
49. Yu, H., Hatzivassiloglou, V.: Owards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (2003)
50. Zhao, W.X., Guo, Y., Yan, R., He, Y., Li, X.: Timeline generation with social attention. In: Proceedings of the 36th International ACM, SIGIR conference on research and development in Information Retrieval, pp. 1061–1064 (2013)
51. Zhou, E., Zhong, N., Li, Y.: Extracting news blog hot topics based on the W2T methodology. World Wide Web **17**(3), 377–404 (2014)