Event Phase Extraction and Summarization

Chengyu Wang[1], Rong Zhang[1], Xiaofeng He[1](✉), Guomin Zhou[2], and Aoying Zhou[1]

[1] Institute for Data Science and Engineering, East China Normal University, Shanghai, China
chywang2013@gmail.com, {rzhang,xfhe,ayzhou}@sei.ecnu.edu.cn
[2] Zhejiang Police College, Hangzhou, Zhejiang, China
zhouguomin@zjjcxy.cn

Abstract. Text summarization aims to generate a single, concise representation for documents. For Web applications, documents related to an event retrieved by search engines usually describe several event phases implicitly, making it difficult for existing approaches to identify, extract and summarize these phases. In this paper, we aim to mine and summarize event phases automatically from a stream of news data on the Web. We model the semantic relations of news via a graph model called *Temporal Content Coherence Graph*. A structural clustering algorithm *EPCluster* is designed to separate news articles corresponding to event phases. After that, we calculate the relevance of news articles based on a vertex-reinforced random walk algorithm and generate event phase summaries in a relevance maximum optimization framework. Experiments on news datasets illustrate the effectiveness of our approach.

Keywords: Event phase summarization · Structural clustering · Vertex-reinforced random walk

1 Introduction

The information overload on the Web motivates the automatic generation of event summaries from documents [1–4] which aims to generate a single, concrete representation of the event. The accuracy and conciseness of summaries are essential for Web applications, such as Web search, news recommendations, etc.

It can be noticed that, existing approaches model an event as one unit and generate a single summary, paying little attention to the fact that there exist several *phases* in long-span, complicated events. Take the case *Egypt Revolution* as an example. Major phases 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]. More recently, the task of timeline generation produces a series of correlated component summaries,

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

W. Cellary et al. (Eds.): WISE 2016, Part I, LNCS 10041, pp. 473–488, 2016.
DOI: 10.1007/978-3-319-48740-3_35

ordered by time [5,6]. However, the entries in a timeline are simply arranged in a sequence, lacking a more structured representation of event phases.

To facilitate deeper analysis on these events, the task we aim to solve in this paper is: *how to automatically extract event phases from a news collection and generate event phase summaries.* It is interesting for several reasons: (i) it groups news articles describing each phase together, instead of considering content similarity only; (ii) it helps readers achieve a better understanding of complicated events by event phase summaries; and (iii) it potentially improves the performance of other tasks such as timeline generation.

To solve the problem, we employ a "divide-and-conquer" method to generate summaries individually after identifying event phases. Because these phases are implicitly expressed in the form of natural language text, we first define two semantic relations (i.e., content coherence and temporal influence) in a news collection via a graph model called *Temporal Content Coherence Graph* (TCCG). A structural clustering algorithm *EPCluster* is designed to extract event phases based on TCCG, in which each phase is represented by a subset of news articles.

For new articles related to a single phase, we design a ranking algorithm based on vertex-reinforced random walk process to calculate the relevance scores of news articles. Based on previous research [6], we model an event phase summary as top-k news headlines and their publication time, and employ a greedy, approximate optimization algorithm to select the corresponding news articles.

In summary, this paper makes the following major contributions:

- We propose and formalize the event phase extraction and summarization problem. A graphical structure TCCG is proposed to model the content coherence and temporal influence relations among news articles.
- A structural clustering algorithm *EPCluster* is designed to group news articles related to the same event phase. We introduce a relevance optimization framework to select top-k news articles to generate event phase summaries.
- We conduct extensive experiments and a case study on news datasets to illustrate the effectiveness of our approach.

The rest of this paper is organized as follows. Section 2 summarizes the related work. We define the event phase extraction and summarization problem formally in Sect. 3. Details of the proposed algorithms are described in Sects. 4 and 5. Experiments are presented in Sect. 6. We conclude our paper and discuss the future work in Sect. 7.

2 Related Work

Given a collection of news articles regarding the same event, various approaches have been proposed to provide users a more concrete representation of the event. Most of the approaches can be classified into two categories: *Multi-Document Summarization* (MDS) and *Timeline Generation* (TG). In this section, we provide an overview of research on these fields.

MDS is a technique of extracting the most salient information from a document collection and transferring it into a brief and informative sentence collection. This problem has been addressed using various paradigms, categorized into two types: extraction-based and abstraction-based. Extraction-based methods assign importance scores to sentences or paragraphs and extract ones with highest scores. Score assignment can be determined by using heuristic and NLP rules, and considering semantic relationships between textual units [1]. There are also some machine learning models for this task. Conroy and O'Leary [2] employ an HMM model to tag important textual units as summaries. More recently, He et al. [3] introduce a sparse coding approach to model each sentence in documents as a linear combination of summary sentence. Additionally, graph-based methods are efficient to rank sentences in documents, such as LexRank [7], cluster-based link analysis [8], etc. Abstraction-based methods utilize the natural language generation technique to create a summary that is closest to the corresponding human-generated summary. In Qian and Liu's work [4], smaller units such as words and phrases are used in the generation process, resulting in more informative summaries.

TG is another research effort to summarize evolutionary news articles by generating component summaries along the timeline. Timelines can also be generated by applying MDS on news articles on each individual date. However, the constraints among temporal components are not modeled in the above approaches [5]. For example, Yan et al. [5] model the trans-temporal characteristics among these component summaries by temporal projection. The headlines of news articles are more informative than the contents, which are exploited by Tran et al. [6] to generate timelines directly via influence-based random walk. Ng et al. [9] construct timelines and incorporate them into an MDS system. It shows that the usage of timelines can improve the performance of MDS.

In summary, both MDS and TG provide concrete information for readers. However, for long-span events, it is necessary to decompose the events into more fine-grained event phases. 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

In this section, we introduce some key concepts used in this paper, formally defining the problem of event phase extraction and summarization.

A news article d_i is a triple, represented as $d_i = (h_i, t_i, s_i)$ where h_i, t_i and s_i denote the headline, the publication time and the sentence collection of text contents. A news collection is a set of news articles $D = \{d_i\}$ where d_i is a news article. In classical aging theory, the life cycle of an event is modeled as four stages: birth, growth, decay and death [10]. However, in a real-life, complicated event, it is difficult to capture the characteristics of the event using only four stages. To overcome this problem, we regard an event as a collection of several event phases. We first introduce the concept of event phase summary as follows:

Definition 1. Event Phase Summary. *An event phase summary P is a collection of k news headline and publication time pairs, denoted as* $P = \{(h_i, t_i)\}_{i=1}^{k}$.

Event phases, however, are unknown before the summarization process, and thus need to be identified beforehand. The task of event phase extraction and summarization is defined as follows:

Definition 2. Event Phase Extraction and Summarization. *Given a news collection D and a positive integer k, the goal is to generate a collection of N event phases* $\mathbf{P} = \{P_j\}_{j=1}^{N}$ *where P_j is an event phase summary, i.e.,* $P_j = \{(h_i, t_i)\}_{i=1}^{k}$.

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

4 Event Phase Extraction

In this section, we present our approach for event phase extraction in detail. The high-level framework is illustrated in Fig. 1.

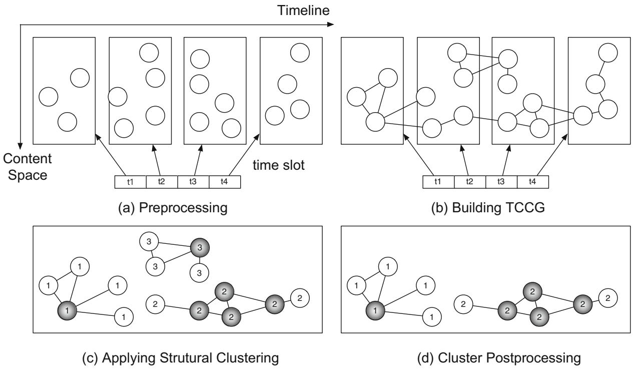
The major challenge is to determine how to measure the degree that two news articles report the same event phase so that they can be grouped into the same cluster. Here, we consider two key factors in terms of content space and time by defining two semantic relations between news articles. Next, the 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, we add an additional postprocessing step 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. However, little has been done to define relations between news articles. In this paper, we study the characteristics of news articles, and introduce two relations, namely content coherence and temporal influence.

Content Coherence. If two news articles are related to the same event phase, they are not necessarily similar in content 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), we define the content

[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.



(a) Preprocessing

(b) Building TCCG

(c) Applying Strutural Clustering

(d) Cluster Postprocessing

Fig. 1. General framework of event phase extraction.

coherence relation considering both *topic level* and *entity level* similarity. We calculate the strength of the relation by a content coherence score, denoted as $w_c(d_i, d_j) \in [0, 1]$.

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* [11]. We regard it as a signal for identifying the change in event phases. To learn the topics, we employ Latent Dirichlet Allocation (LDA), a well-established topic model for documents [12]. For each news article $d_i \in D$, LDA associates it with a topic distribution vector $\boldsymbol{\theta_i}$. For two articles d_i and d_j, the difference between topic distributions are captured by Jansen-Shannon divergence, defined as:

$$D_{JS}(\boldsymbol{\theta_i} \| \boldsymbol{\theta_j}) = \frac{D_{KL}(\boldsymbol{\theta_i} \| \overline{\boldsymbol{\theta}}) + D_{KL}(\boldsymbol{\theta_j} \| \overline{\boldsymbol{\theta}})}{2} \tag{1}$$

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

Another observation is that, entities (e.g. people, locations and organizations) play a vital role in news reports. 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 techniques. Instead, we utilize our *NERank* algorithm [13] to extract key entities in the news collection D, denoted as E_D. Let $\boldsymbol{c_i}$ 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:

$$TC(\boldsymbol{c_i}, \boldsymbol{c_j}) = \frac{\boldsymbol{c_i^T} \cdot \boldsymbol{c_j}}{||\boldsymbol{c_i}||^2 + ||\boldsymbol{c_j}||^2 - \boldsymbol{c_i^T} \cdot \boldsymbol{c_j}} \tag{2}$$

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

$$w_c(d_i, d_j) = \alpha \cdot (1 - D_{JS}(\boldsymbol{\theta_i} \| \boldsymbol{\theta_j})) + \beta \cdot TC(\boldsymbol{c_i}, \boldsymbol{c_j}) \tag{3}$$

where α and β are tuning parameters that control the strength of entity level and topic level similarity measures. We require α, $\beta \in [0, 1]$ and $\alpha + \beta = 1$. For simplicity, we set $\alpha = \beta = \frac{1}{2}$ in this paper and leave automatic learning for future research.

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 to the phenomenon that if publication time of d_i and d_j are close, they are likely to report the same event phase and vice versa. 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 $\Gamma(\cdot)$ [14] to map $\Delta t_{i,j}$ to a real number in $[0, 1]$:

$$\Gamma(\Delta t_{i,j}) = \begin{cases} \frac{1}{2}(1 + \cos(\frac{\Delta t_{i,j} \cdot \pi}{\sigma})) & (\Delta t_{i,j} \leq \sigma) \\ 0 & (\Delta t_{i,j} > \sigma) \end{cases} \tag{4}$$

where σ 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]

4.2 *EPCluster*: A Structural News Clustering Algorithm

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

TCCG. 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 we can determine there is a strong semantic relation between d_i and d_j. In this paper, we introduce two parameters μ_1 and μ_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 Fig. 1(b). Here, we formally define the concept of TCCG as follows:

Definition 3. Temporal Content Coherence Graph. *A Temporal Content Coherence Graph w.r.t. parameters μ_1 and μ_2 and news collection D is an undirected graph $G_D = (V, E)$ such that:*

[3] In the implementation, we set one day as a time slot and compute $w_t(\cdot)$ based on publication date difference. See Fig. 1(a) and (b).

- *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]

***EPCluster* Algorithm.** Structural clustering has been extensively exploited to summarize and analyze various types of networks [15]. 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.

While traditional structural clustering algorithm *SCAN* [15] requires two parameters, *EPCluster* takes three parameters as input, namely μ_1, μ_2 and $MinPts$, where μ_1 and μ_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 μ_1 and μ_2 similarity of an object. Here, we first define the concept of (μ_1, μ_2)-neighborhood:

Definition 4. (μ_1, μ_2)-Neighborhood. *The (μ_1, μ_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 of $w_c(d_i, d_j) > \mu_1$ 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 (μ_1, μ_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.*

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.*

In the algorithm, with the TCCG constructed, it 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.

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 of $O(|E|)$. Therefore, *EPCluster* is an algorithm of which the complexity is linear in terms of edges.

[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.

Algorithm 1. *EPCluster* Algorithm

```
Input: News collection $D$, parameters $\mu_1, \mu_2, MinPts$.
Output: Cluster collection $\mathbf{C}$.
$\mathbf{C} = \emptyset$, $clusterID = 1$;
for each $d_i \in D$ do
    if $d_i$ is not visited then
        $N(d_i) =$SearchNeighbors$(d_i, \mu_1, \mu_2)$;
        if $|N(d_i)| \geq MinPts$ then
            $C_{clusterID}$=ExpandCluster$(d_i, \mu_1, \mu_2, MinPts)$;
            $\mathbf{C} = \mathbf{C} \cup \{C_{clusterID}\}$;
            $clusterID = clusterID + 1$;
        end if
    end if
end for
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:

Article Quantity. For cluster $C_i \in \mathbf{C}$, denote $N(C_i) = \frac{|C_i|}{|D|} \times 100\,\%$ as the percentage of articles in that cluster.

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 $T(C_i) = t_{max}^i - t_0^i$ where

$$t_0^i = \max\{t_1^i, t_{Q1}^i - 1.5 \cdot |t_{Q3}^i - t_{Q1}^i|\} \tag{5}$$

$$t_{max}^i = \min\{t_{|C_i|}^i, t_{Q3}^i + 1.5 \cdot |t_{Q3}^i - t_{Q1}^i|\} \tag{6}$$

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

$$ATS(C_i) = 1 - \frac{2\sum_{d_m, d_n \in C_i (m<n)} D_{JS}(\boldsymbol{\theta_m} \| \boldsymbol{\theta_n})}{|C_i| \cdot (|C_i| - 1)} \tag{7}$$

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

$$AES(C_i) = \frac{2\sum_{d_m, d_n \in C_i (m<n)} TC(\boldsymbol{c_m}, \boldsymbol{c_n})}{|C_i| \cdot (|C_i| - 1)} \tag{8}$$

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 $\boldsymbol{w}$ gives different weights for each feature in $F(C_i)$. Therefore, for each cluster C_i, we define a score function $Score(C_i) = \boldsymbol{w} \cdot F(C_i)$ 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:

$$f(C_i) = \frac{1}{1 + e^{-\boldsymbol{w} \cdot F(C_i)}} \tag{9}$$

We learn the weight vector $\boldsymbol{w}$ 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 extraction 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 where $d_j \in V_{C_i}$ iff $d_j \in C_i$ and $(d_j, d_k) \in E_{C_i}$ iff $d_j \in C_i$, $d_k \in C_i$ and $(d_j, d_k) \in E$. Refer to a simle example in Fig. 1(d).

While the standard PageRank algorithm [16] 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*. 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 [17,18] to balance *centrality* and *diversity* in ranking.

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

In our 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 $\boldsymbol{R_0}$ as a $|C_i| \times 1$ prior ranking vector for articles in C_i. Without prior knowledge, $\boldsymbol{R_0}$ is set uniformly, i.e., $\boldsymbol{R_0} = \frac{1}{|C_i|} \boldsymbol{e}$ where $\boldsymbol{e}$ is a $|C_i| \times 1$ vector with all elements assigned to 1. $M^{(0)}_{m,n}$ (the element in the mth row and nth column of the prior transition matrix $\mathbf{M_0}$) is defined using the fusion of the two relation strength scores:

$$M^{(0)}_{m,n} = \begin{cases} \frac{1}{Z} \cdot w_c(d_m, d_n) \cdot w_t(d_m, d_n) & (d_m, d_n) \in E_{C_i} \\ 0 & otherwise \end{cases} \tag{10}$$

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

$$\boldsymbol{M_{n+1}} = \lambda \boldsymbol{T_n} \cdot \boldsymbol{M_n} + (1 - \lambda)\boldsymbol{M_0} \tag{11}$$

where $\boldsymbol{T_n} = [\boldsymbol{R_n}\boldsymbol{R_n} \cdots \boldsymbol{R_n}]$ 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:

$$\boldsymbol{R_{n+1}} = \lambda \boldsymbol{M_{n+1}} \cdot \boldsymbol{R_n} + (1 - \lambda)\boldsymbol{R_0} \tag{12}$$

The above iterative formula defines an ergodic random walk process in a Markov chain. As shown in [18], it also converges to a stationary distribution. After sufficient large times of iteration N^*, we obtain $r(d_j) = \sum_{d_k \in C_i} M^{(n)}_{j,k} \cdot r(d_k)$ as the relevance score of d_j when $n > N^*$.

Algorithm 2. News Article Ranking Algorithm

```
Input: News cluster $C_i$, parameter $\lambda$.
Output: Ranking vector $\boldsymbol{R}$.
Compute $\boldsymbol{M}$ based on $C_i$;
$\boldsymbol{R_0} = \frac{1}{|C_i|}\boldsymbol{e}$, $\boldsymbol{M_0} = \boldsymbol{M}$, $n = 0$;
while not converge do
    $\boldsymbol{T_n} = [\boldsymbol{R_n}\boldsymbol{R_n} \cdots \boldsymbol{R_n}]$;
    $\boldsymbol{M_{n+1}} = \lambda \boldsymbol{T_n} \cdot \boldsymbol{M_n} + (1 - \lambda)\boldsymbol{M_0}$;
    $\boldsymbol{R_{n+1}} = \lambda \boldsymbol{M_{n+1}} \cdot \boldsymbol{R_n} + (1 - \lambda)\boldsymbol{R_0}$;
    $n = n + 1$;
end while
return $\boldsymbol{R} = \boldsymbol{R_n}$;
```

5.2 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, approximate 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 select news articles d_m and d_n, we require $w_c(d_m, d_n) \leq \mu_1$ and $w_t(d_m, d_n) \leq \mu_2$. Here, we present our *News Selection* optimization problem:

$$\begin{aligned} \max_{S_i \subset C_i} \quad & R(S_i) = \sum_{d_j \in S_i} r(d_j) \\ \text{subject to} \quad & |S_i| = k \\ & w_c(d_m, d_n) \leq \mu_1, w_t(d_m, d_n) \leq \mu_2, \forall d_m, d_n \in S_i \end{aligned} \tag{13}$$

The proposed optimization problem can be seen as a special case of the *budgeted maximum coverage problem* [19], 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. [19]. 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 and compare it with baselines. All the codes are written in JAVA, and run on a PC with an Intel CPU 2.9 GHz and 16 GB memory.

6.1 Datasets

The news datasets we used in this paper are publicly available from [6], which 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 the Google search engine. The detailed statistics are illustrated in Table 1.

Table 1. Summary of datasets.

Dataset	Event	#Article	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 Evaluation on Event Phase Extraction

Experimental Settings. To our knowledge, there is no prior work regrading event phase extraction. 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 [20]. For each dataset D_i, we randomly generate news article pairs, denoted as $T_i = \{(d_m, d_n)\}$. 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:

$$Precision(T_i) = \frac{|\{(d_m, d_n) \in T_i | v_{m,n} = 1 \wedge v^{'}_{m,n} = 1\}|}{|\{(d_m, d_n) \in T_i | v^{'}_{m,n} = 1\}|} \tag{14}$$

$$Recall(T_i) = \frac{|\{(d_m, d_n) \in T_i | v_{m,n} = 1 \wedge v^{'}_{m,n} = 1\}|}{|\{(d_m, d_n) \in T_i | v_{m,n} = 1\}|} \tag{15}$$

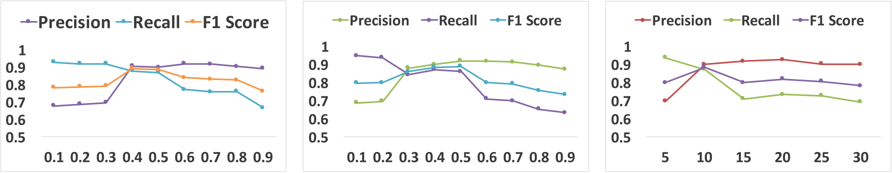
$$F1\ Score(T_i) = \frac{2 \cdot Precision(T_i) \cdot Recall(T_i)}{Precision(T_i) + Recall(T_i)} \tag{16}$$

In total, we have 300 labeled new article pairs for each dataset. We report macro-average precision, recall and F1 score in the following experiments.

Parameter Tuning. We tune three parameters in *EPCluster*, namely μ_1, μ_2 and $MinPts$. We fix two parameters and vary the remaining one at each time. The results are illustrated in Fig. 2. It can be seen that when $\mu_1 = 0.4$, $\mu_2 = 0.5$ and $MinPts = 10$, *EPCluster* achieve the best results.

Method Comparison. While document clustering is a well-studied problem, we compare our method with classical approaches and the variant of our method, introduced as follows:

- **VSMCluster** - KMeans using word features of TF-IDF weights.
- **TopicCluster** - KMeans using topic distributions based on LDA [12].
- **SCAN** [15] - structural clustering algorithm for network partitioning.
- **EPCluster-C** - our *EPCluster* algorithm without postprocessing.



(a) Varying μ_1 (b) Varying μ_2 (c) Varying $MinPts$

Fig. 2. Clustering results of *EPCluster* under different parameter settings.

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. To compare our method with the state-of-the-art structural clustering algorithm *SCAN* [15], we first construct a TCCG and then apply *SCAN* on the graph. The results are presented in Table 2.

Table 2. Experimental results of event phase extraction.

Method	VSMCluster	TopicCluster	SCAN	EPCluster-C	Our method
Precision	0.35	0.52	0.78	0.81	**0.89**
Recall	0.74	0.67	0.72	**0.79**	0.78
F1 score	0.48	0.59	0.75	0.80	**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 and thus are similar in content, these methods are not suitable for clustering-based event phase extraction method. *SCAN* algorithm has a relative 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-C* and our method shows that the postprocessing step is effective to improve the performance of event phase extraction.

6.3 Evaluation on Event Phase Summarization

Ground Truth. The ROUGE framework [21] has been extensively used to evaluate the effectiveness of document summarization. However, the summaries we generate are headlines, rather than documents. Tran et al. [6] previously propose a headline summary evaluation framework based on the relevance of machine-generated timelines compared with ground truth timelines. In this paper, we obtain the timeline summaries 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 [6].

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.** [6] - the timeline generation method especially for headlines.
- **Chieu et al.** - [22] a timeline generation method based sentence popularity.
- **Our Method (PageRank)** - the variant of our approach which adopts simple PageRank method for relevance calculation.

We also consider the following two benchmark methods:

- **Random** - selects k news articles randomly.
- **Longest** - selects top-k longest headlines due to the informativeness.

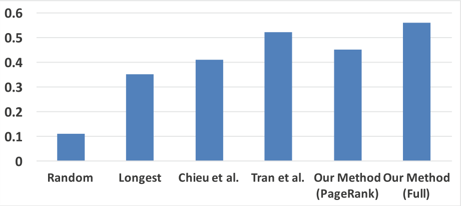


Fig. 3. Relevance evaluation of event phase summarization.

Experiments and Results. To evaluate these methods, we extract generated summaries from 106 dates that are appeared in the ground truth summaries. We present the ground truth and machine-generated summaries to human annotators and ask them to label each headline as relevant or not. We take the average relevance scores for each method as the evolution metrics. The results are presented in Fig. 3. It can be seen that the results of benchmark approaches are not as good as others because they lack textual analysis on news articles. 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. 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. 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.

[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.

Case Study. We present the event phase summaries of *Egypt Revolution* produced by our approach. Due to space limitation, we only present the publication dates and headlines of two news articles in each event phase. We also manually add a brief description for each phase, shown in Table 3. It shows that our approach can identify and summarize fine-grained event phases effectively.

Table 3. 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
2011.2.11	Hosni Mubarak resigns and Egypt celebrates a new dawn
Event Phase #2 *Egypt under the Rule of Military Power*	
2011.4.9	Egyptian soldiers attack Tahrir Square protesters
2011.7.10	Protests spread in Egypt as discontent with military rule grows
Event Phase #3 *Mohammed Morsi Won Presidential Election*	
2012.5.23	First round of presidential election
2012.6.24	Election officials declare Morsi the winner
Event Phase #4 *Protest against Morsi and Muslim Brotherhood*	
2013.1.27	Egypt's Mohammed Morsi declares state of emergency, imposes curfew
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 Presiden
2013.7.6	Morsi's ouster in Egypt sends chill through political Islam

7 Conclusion and Future Work

In this paper, we formalize the problem of event phase extraction and summarization. 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 summaries by relevance maximum optimization. Experiments show that our method can solve the problem effectively. In the future, we will focus on improving the performance of MDS and TG when event phases are considered.

Acknowledgements. This work is partially supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904, Shanghai Agriculture Science Program (2015) Number 3-2 and NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant No. U1509219.

References

1. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: SIGIR, pp. 19–25 (2001)
2. Conroy, J.M., O'Leary, D.P.: Text summarization via hidden markov models. In: SIGIR, pp. 406–407 (2001)
3. He, Z., Chen, C., Bu, J., Wang, C., Zhang, L., Cai, D., He, X.: Document summarization based on data reconstruction. In: AAAI (2012)
4. Qian, X., Liu, Y.: Fast joint compression and summarization via graph cuts. In: EMNLP, pp. 1492–1502 (2013)
5. Yan, R., Kong, L., Huang, C., Wan, X., Li, X., Zhang, Y.: Timeline generation through evolutionary trans-temporal summarization. In: EMNLP, pp. 433–443 (2011)
6. Tran, G., Alrifai, M., Herder, E.: Timeline summarization from relevant headlines. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 245–256. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16354-3_26
7. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Qiqihar Junior Teachers Coll. **22**, 2004 (2011)
8. Wan, X., Yang, J.: Multi-document summarization using cluster-based link analysis. In: SIGIR, pp. 299–306 (2008)
9. Ng, J., Chen, Y., Kan, M., Li, Z.: Exploiting timelines to enhance multi-document summarization. In: ACL, pp. 923–933 (2014)
10. Chen, C.C., Chen, Y.-T., Sun, Y., Chen, M.C.: Life cycle modeling of news events using aging theory. In: Lavrač, N., Gamberger, D., Blockeel, H., Todorovski, L. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 47–59. Springer, Heidelberg (2003). doi:10.1007/978-3-540-39857-8_7
11. Knights, D., Mozer, M.C., Nicolov, N.: Detecting topic drift with compound topic models. In: ICWSM (2009)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
13. Wang, C., Zhang, R., He, X., Zhou, A.: Nerank: ranking named entities in document collections. In: WWW, pp. 123–124 (2016)
14. De Kretser, O., Moffat, A.: Effective document presentation with a locality-based similarity heuristic. In: SIGIR, pp. 113–120 (1999)
15. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: SCAN: a structural clustering algorithm for networks. In: KDD, pp. 824–833 (2007)
16. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. **30**(1–7), 107–117 (1998)
17. Pemantle, R.: Vertex-reinforced random walk. Probab. Theory Relat. Fields **92**(1), 117–136 (1992)
18. Mei, Q., Guo, J., Radev, D.R.: Divrank: the interplay of prestige and diversity in information networks. In: KDD, pp. 1009–1018 (2010)
19. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Inf. Process. Lett. **70**(1), 39–45 (1999)
20. Chen, J., Niu, Z., Fu, H.: A multi-news timeline summarization algorithm based on aging theory. In: Cheng, R., Cui, B., Zhang, Z., Cai, R., Xu, J. (eds.) APWeb 2015. LNCS, vol. 9313, pp. 449–460. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25255-1_37
21. Lin, C., Hovy, E.H.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: HLT-NAACL (2003)
22. Chieu, H.L., Lee, Y.K.: Query based event extraction along a timeline. In: SIGIR, pp. 425–432 (2004)