applied sciences

MDPI

Article

Key News Event Detection and Event Context Using Graphic Convolution, Clustering, and Summarizing Methods

Zheng Liu [1,2,3,†], Yu Zhang [1,3,*,†], Yimeng Li [1,3] and Chaomurilige [1,3,*]

[1] School of Information Engineering, Minzu University of China, Beijing 100081, China; 21400174@muc.edu.cn (Z.L.)
[2] School of Chinese Ethnic Minority Languages and Literatures, Minzu University of China, Beijing 100081, China
[3] Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance, Ministry of Education, Minzu University of China, Beijing 100081, China
* Correspondence: 21302041@muc.edu.cn (Y.Z.); chaomurilige@muc.edu.cn (C.)
† These authors contributed equally to this work.

Abstract: The detection of key events and identification of the events' context have been widely studied to detect key events from large volumes of online news and identify trends in such events. In this paper, we propose a Key News Event Detection and Context Method based on graphic convolving, clustering, and summarizing methods. Our method has three main contributions: (1) We propose the use of position vectors as time-embedding feature representations and concatenate semantic and time-embedding features as node features of the graph to distinguish different nodes of the graph. Additionally, a temporal nonlinear function was constructed using time embedding to objectively describe the effect of time on the degree of association between nodes. (2) We update the graph nodes using a graph convolutional neural network to extract deep semantic information about individual nodes of a high-quality phrase graph, thereby improving the clustering capability of graph-based key event detection. (3) We apply a summary generation algorithm to a subset of news data for each key event. Lastly, we validated the effectiveness of our proposed method by applying it to the 2014 Ebola dataset. The experimental results indicate that our proposed method can effectively detect key events from news documents with high precision and completeness while naturally generating the event context of key events, as compared to EvMine and other existing methods.

Keywords: key event; event context; graphic convolving; clustering; summarizing

check for updates

Citation: Liu, Z.; Zhang, Y.; Li, Y.; Chaomurilige. Key News Event Detection and Event Context Using Graphic Convolution, Clustering, and Summarizing Methods. *Appl. Sci.* **2023**, *13*, 5510. https://doi.org/10.3390/app13095510

Academic Editor: Dimitris Mourtzis

Received: 31 March 2023
Revised: 12 April 2023
Accepted: 24 April 2023
Published: 28 April 2023

1. Introduction

The aim of so-called key event detection is to obtain a set of clusters in a corpus about a certain topic, with each cluster containing a set of articles describing the same event. The event context is a process of generating event descriptions for each key event and arranging them by time of occurrence. With the rapid development of computer and network technology, an exponentially growing, complex, and massive amount of online data and information has emerged. As a result, people often only see partial information, and it is thus difficult to obtain a comprehensive and quick overview of the big picture. Key event detection can be involved in applications across different modalities, such as video summarization [1] and augmented reality [2]. Accordingly, the search for more advanced techniques to detect events and their contexts, enabling the rapid and accurate detection of key events in large-scale data and the timely observation of the context of such key events [3], has become a growing subject for many researchers.

News events evolve rapidly and are variable in nature [4]. It is therefore difficult to obtain large amounts of annotated data. Meanwhile, the models obtained on the basis of training data are less portable. Thus, the use of unsupervised methods such as clustering for the detection of key events has also become a trend. For example, Yang et al. [5] constructed a KeyGraph and then used a community discovery algorithm for clustering. A Burst Information Network (BINet) based event detection model, the Time-aware Hierarchical Bayesian Model (TaHBM), was proposed by Ge et al. [6]. Zhang et al. proposed an unsupervised key event detection model, EvMine [7]. Both of the above methods incorporate time information. Usually, a key event is discussed in articles with similar content and is published within a certain timeframe. After an important news event occurs, it is widely discussed in the following days, but the discussion gradually fades as the time frame expands. This shows that time information can serve as an indicative factor for describing different key events in news. Integrating time information into clustering algorithms can improve their clustering effectiveness. Typically, a news article includes not only the text of the news itself, but also the time of publication. This publishing pattern provides a natural convenience for utilizing time information. Among the aforementioned models, EvMine is a more effective method for event detection and has been successfully applied in detecting key events. However, EvMine has the following limitations: (1) The practice of artificially assigning weights to edges between graph nodes based on experience is highly subjective and lacks generalizability. (2) The deep semantic information contained in the nodes of the original peak phrase graph is not considered. The nodes are clustered in the original peak phrase graph, and the clustering results are often unsatisfactory. (3) As EvMine focuses on key event detection, it lacks automatic identification of the key event context. As a result, there is a need to improve the utilization of time information in constructing the connected edges of a graph to obtain high-quality semantic feature information for the nodes and enable automatic identification of the context of key events. These three key points are crucial for the study of key event detection and event context.

To address the above issues, inspired by the EvMine model, this paper proposes a method for detection of key events in news documents and identification of the events' context based on graphic convolving, clustering, and summarizing (GCS). The work presented in this paper mainly includes: (1) The union set of the semantic embedding of high-quality phrases extracted from the news corpus and the embedded release times of the documents to which the phrases belong are used as the nodes of the graph. The nodes are connected based on the news release rules in reality, thereby constructing an initial graph of high-quality phrases. (2) A graph convolutional network (GCN) is applied to high-quality phrase graphs to obtain deep semantic features of each node. (3) The advanced high-quality phrase graph consists of the semantic relatedness between two different nodes and the weights of the edges updated with the temporal relatedness between the two nodes described by a nonlinear function. (4) Nodes of advanced high-quality phrase graphs are clustered using the Louvain community discovery algorithm to classify news documents into subsets of news data for different key events. (5) The summary generation algorithm is applied to each subset of key event news data to generate summary descriptions of key events. The summary descriptions are then automatically ranked based on the span of news releases in the key event news. Finally, the event context of the events is obtained.

2. Related Work

According to the format of input news text, the key event detection task can be divided into sentence-level event detection and document-level key event detection. For sentence-level event detection tasks, using graph structures has become a trend. Xie et al. [8] proposed a graph parsing method using a dependency-tree-based graph and convolutional neural networks. He et al. [9] presented a self-constructed dependency and graph convolution network for event detection. For document-level key event detection, the objective of certain recent studies is to identify narratives within a broad collection of news articles. Saravanakumar et al. [10] proposed a hierarchical clustering algorithm that leverages both textual and contextual features to group similar news stories together. Sia et al. [11] discussed the use of clusters of pre-trained word embeddings for fast and effective topic modeling. Santos et al. [12] presented a method for simplifying multilingual news clustering through projection from a shared space. The trend of using graph structures has also begun to become popular in document-level key event detection. Yang et al. [5] first constructed a KeyGraph based on textual keyword co-occurrence relationships, then combined this with a community detection algorithm to segment the KeyGraph and used the extracted topic features to detect events in social media streams. S. Gaglio et al. [13] performed frequent pattern mining based on keywords, using sets of words that frequently occur together as events. Ge et al. [6] proposed a Burst Information Network (BINet) based event detection model, the Time-aware Hierarchical Bayesian Model (TaHBM), which first identifies key nodes or key regions in a BINet as the center of mass and constructs clusters, thus transforming the event detection problem into a community detection implementation on a BINet network. Brochier et al. [14] proposed document network embedding with topic word attention. One of the most commonly used community discovery algorithms is the Louvain algorithm [15], which is known for its speed and scalability, making it particularly useful for large-scale networks. Zhang et al. [7] proposed an unsupervised key event detection model, EvMine. Furthermore, the integration of time information has also become a research direction in these methods. Ge et al. [6] incorporated time information into the Bayesian model, resulting in an improvement in performance. Zhang et al. [7] constructed a peak phrase graph through statistical analysis of time information, which improved the effectiveness of document clustering in subsequent stages.

Among these research methods, the selection of keywords is particularly important. Guille A et al. [16] proposed a classical keyword generation method. Chen et al. [17] proposed a topic model that uses the lexical relevance of words in a given dictionary to extract coherent keywords for a topic. Due to the broad applicability of phrases to text-related tasks, it is often used as a form of expression for keywords. Some researchers [18] used an N-Gram for event detection, but this also invokes some nonsensical collocations and requires specific methods to select semantic phrases. Gu et al. [19] proposed a new unsupervised context-aware quality phrase tagger, UCPhrase. In the unsupervised case, the context of each N-Gram was viewed to find the largest pattern and to be able to find more complete phrases that served as better supervised signals for the feature-based classifier for the UCPhrase context. A pre-trained language model with a transformer containing an attention mechanism is then used to identify high-quality phrases in new sentences.

However, the semantic feature extraction of keywords determines the accuracy of the clustering results. Microsoft Palangi et al. [20] proposed a text matching model based on a long short-term memory recurrent neural network (LSTM-RNN), which transforms query words and target documents into vectors and calculates cosine similarity. Zhang and Liu et al. [21] implemented a bi-directional LSTM by proposing a sentence-state-based LSTM. It maintains a global structured sentence-level semantic vector that is used and updated by each recurrent unit and can better capture dependencies based on global information. To more rationally describe the semantic features of long texts, by combining graph representations with neural networks, graph convolutional neural networks [22] were generated. Yao et al. [23] proposed the TextGCN model to learn word and document semantic representations and perform text classification without pretraining based on word co-occurrence matrices and graph convolution of heterogeneous semantic association graphs generated from document–word associations.

3. Methodology

Figure 1 shows the general structure of the GCS proposed in this document, which consists mainly of the following parts: (1) graph construction based on high-quality phrases; (2) graph update based on GCN; (3) graph-based key events detection; and (4) key event context generation. The content of each part is described in detail below.

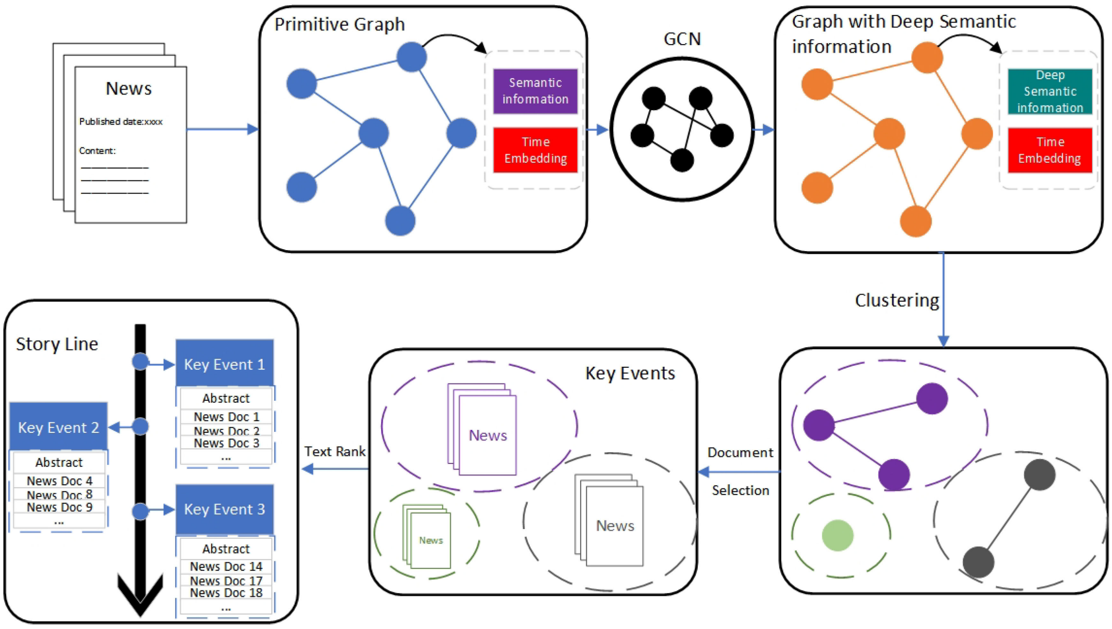


Figure 1. The overall architecture of Key Event Detection from news documents and Event Context.

3.1. Graph Construction Based on High-Quality Phrases

3.1.1. Semantic Embedding of High-Quality Phrases

Given a corpus $D = \{d_1, d_2, \ldots, d_n\}$, where d_i is the ith document, $i \in [1, n]$, and n is the number of documents in the corpus, a classifier $f(.;\theta)$ is trained using UCPhrase to extract high-quality phrases from each document:

$$HP_i = f(d_i;\theta) \tag{1}$$

where $HP_i = \{hp_1^i, hp_2^i, \ldots\}$ is the set of high-quality phrases in the ith document, and hp_j^i is the jth high-quality phrase in the ith document, $hp_j^i \neq hp_k^i, k \neq j$. The number of high-quality phrases per document is variable.

Suppose $DHP = \{HP_1, HP_2, \ldots, HP_n\}_M$ is the set of high-quality phrases in the entire documents. M is the total number of high-quality phrases extracted:

$$DHP = \overset{n}{\underset{i=1}{\cup}} HP_i \tag{2}$$

Assume that hp_j^i high-quality phrase appears N_p times in a different document, i.e., $\underbrace{hp_j^i = hp_k^m = \cdots = hp_g^v}_{N_p}, m \neq i \neq \cdots \neq v$. The semantic embedding of the high-quality phrase itself is denoted by shp_j^i. The semantic embedding of high-quality phrase contexts is denoted by mhp_j^i. Based on this annotated form, the semantic embedding of the high-quality phrases hp_k^m and hp_g^v is denoted as shp_k^m, mhp_k^m, shp_g^v, mhp_g^v. Then the final semantic embedding ehp_j^i of hp_j^i is given by [7]:

$$ehp_j^i = \frac{1}{N_p}\left\{\left[shp_j^i, mhp_j^i\right] + [shp_k^m, mhp_k^m] + \cdots + \left[shp_g^v, mhp_g^v\right]\right\} \tag{3}$$

Using Equation (3) to semantically embed each high-quality phrase in the set DHP, to obtain the set of semantic embeddings of high-quality phrases $EDHP = \{EHP_1, EHP_2, \ldots, EHP_n\}_M$:

$$EDHP = \overset{n}{\underset{i=1}{\cup}} EHP_i \tag{4}$$

where $EHP_i = \{ehp_1^i, ehp_2^i, \ldots\}$ is the set of semantic embeddings of high-quality phrases of the ith document.

3.1.2. Time Embedding of High-Quality Phrases

As the same phrases may appear at different times, temporal information needs to be used to distinguish these nodes at different points in time. In news articles, a key event is usually discussed in a number of articles that are similar in content and are published within a certain time frame. After a key news event has occurred, it is widely discussed in the days that follow, but after the time frame has been extended further, the discussion fades. Therefore, this paper proposes a time-embedding approach for news articles, which can be used to express the relative relationship between the publication times of articles. As temporal information is also simultaneously spatial, time can be understood using the concept of relative position of time. Inspired by positional coding [24], the use of positional information to encode time allows the expression of relative positional information of time.

Given that there are news articles containing high-quality phrases hp_j^i for k days in the corpus, and the vector corresponding to the publication time of a news document on a certain day is the d-dimensional time embedding $t_{index},$, $index \in [1, k]$, then the time embedding corresponding to the earliest release time is t_1 and the time embedding corresponding to the latest release time is t_k. So, the time embedding is given by:

$$t_{index}^l hp_j^i = \begin{cases} \sin\left(\frac{index}{10000^{\frac{l}{d}}}\right), & \text{if } l \text{ is even} \\ \cos\left(\frac{index}{10000^{\frac{l-1}{d}}}\right), & \text{if } l \text{ is odd} \end{cases} \tag{5}$$

where l is the lth element in the time-embedding vector, $l \in [1, d]$.

3.1.3. Graph Construction of High-Quality Phrases

Using Equations (3)–(5), the semantic embedding of each phrase in the set of high-quality phrase semantic embedding and the corresponding news release time embedding of the article in which the phrase is embedded are concatenated as the composite high-quality phrase semantic embedding features, i.e.,

$$\mathrm{te}hp_j^i = \left[ehp_j^i, t_{index}^l hp_j^i\right] \tag{6}$$

The final set of composite high-quality phrase semantic embedding $TEDHP = \{TEHP_1, TEHP_2, \ldots, TEHP_n\}_M$ is formed:

$$TEDHP = \bigcup_{i=1}^{n} TEHP_i \tag{7}$$

where $TEHP_i = \{tehp_1^i, tehp_2^i, \ldots\}$ is the set of semantic embedding of compound high-quality phrases for the ith document.

Using each compound high-quality phrase semantic embedding in the $TEDHP$ as a node feature of the graph, the constructed high-quality phrase graph should have M nodes.

According to the general pattern of news releases in reality: (1) The same key event is usually reported by several media during the same day. So, if the news documents to which the nodes belong have the same publication date, there is an edge between the nodes. (2) News describing a key event breaks on the first day and is reported in the following days. So, if the same phrase appears on consecutive days, there is an edge between these nodes. For node-to-node associations, the high-quality phrase graph is constructed as an undirected graph based on this general pattern of news releases. Meanwhile, if there are edges between two nodes, the initial weights of the edges are both set equal to 1. Figure 2 shows this.

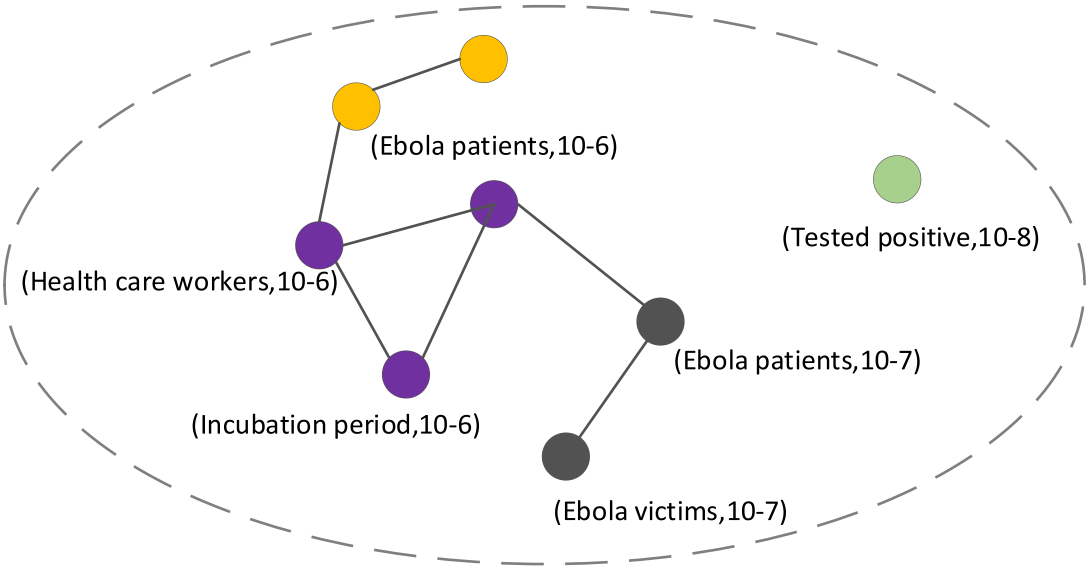


Figure 2. High-quality phrase graph. The time in the figure represents the publication time of the news document where the phrase was found, and the time range is determined by the corpus used, from the earliest published news in the corpus to the latest.

3.2. Graph Optimization

3.2.1. GCN-Based Graph Node Optimization

The high-quality phrase graphs that are currently being constructed have nodes that only consider shallow semantic information and their representation ability is not strong. The graph nodes need to be updated by a graph convolutional neural network to extract deep semantic information. This operation facilitates the clustering or subsequent processing of the graph.

First the nodes of the high-quality phrase graph are denoted as $\{n_1, n_2, \ldots, n_M\}$, where M is the number of nodes. The node number is matched with the semantic embedding of the compound high-quality phrase in $TEDHP = \{TEHP_1, TEHP_2, \ldots, TEHP_n\}_M$, i.e., n_i is the ith compound high-quality phrase semantic embedding, including the phrase embedding and the corresponding time-embedding feature t_i. Let the high-quality phrase graph be $G = (N, E)$, where $N = \{n_1, n_2, \ldots, n_M\}$ is the set of nodes in G and E is the set of edges in G. When performing graph convolution on $G = (N, E)$, the time embedding in node n_i is not considered, i.e., another graph $\hat{G} = \left(\hat{N}, E\right)$ is constructed, where $\hat{N} = \{n_1 - t_1, n_2 - t_2, \ldots, n_M - t_m\}$. A graph convolution update is performed on $\hat{G} = \left(\hat{N}, E\right)$ to extract the deep semantic features of the nodes and obtain a high-quality phrase graph with deep semantic features. The update process is given by Equation (8) [22]:

$$\hat{G}^{l+1} = Relu\left(D^{-\frac{1}{2}}\widetilde{A}D^{-\frac{1}{2}}\hat{G}^{l}W^{l}\right) \tag{8}$$

where, for a GCN with L layers, $l \in [1, L]$, $\hat{G}^l$ is the state at graph input, and $\hat{G}^{l+1}$ is the output at layer l. D is the degree matrix of the graph $\hat{G}^l$, $\widetilde{A} = A + I$ is the adjacency matrix with added self-connections, and A is the adjacency matrix of $\hat{G}^l$. I is the identity array. $Relu$ is the activation function, and W is a parameter in the graph convolutional neural network.

After the GCN convolution operation, a new high-quality phrase graph $G' = \left(\hat{N}', E\right)$ is obtained, where $\hat{N}' = \{\hat{n}_1 + t_1, \hat{n}_2 + t_2, \ldots, \hat{n}_M + t_M\}$, and $\hat{n}_i$ denotes the deep semantic embedding feature of the ith node in the new high-quality phrase graph.

3.2.2. Optimizing Graph Edges

When constructing the initial high-quality phrase graph, the weights of the edges between the nodes are simply set to 1. In practice, however, the weights of the edges between the nodes are closely related to the following factors: (1) the cosine similarity of node-to-node phrase embedding and (2) the influence of time between nodes. A key event in a news article is usually discussed in large bursts by a number of articles, and the heat of that key event decreases more rapidly over time. However, the effect of time information on the nodes is not presented as a conventional linear relationship. Therefore, this paper constructs a nonlinear exponential function $I(u,v)$ to simulate this law, as shown in Equation (9).

$$I(u,v) = e^{(1+CosSim(t_u,t_v))} \tag{9}$$

where $CosSim$ denotes cosine similarity, $u \in [1,M], v \in [1,M]$ denotes two nodes in the graph, and t_u, t_v is the corresponding time embedding.

The weights of the edges are closely related to the cosine similarity of the phrase embedding between nodes, in addition to the influence of time between nodes. Therefore, taking these factors into account, the cosine similarity of phrase embedding between nodes and Equation (9) are combined to design the calculation of edge weights $w_{u,v}$ in high-quality phrase graphs, as shown in Equation (10).

$$w_{u,v} = I(u,v) * \max(CosSim(\hat{n}_u, \hat{n}_v), 0) \tag{10}$$

where $\hat{n}_u, \hat{n}_v$ are the deep semantic embedding features corresponding to the two nodes u, v.

3.3. Key Event Detection

Graph-Based Community Discovery: Since the phrases are extracted from a massive amount of news text, their quantity is also quite large. As a result, the constructed graph structure is also relatively large. Based on the entire process, an unsupervised community detection algorithm needs to be selected. Additionally, it is hoped that the algorithm can perform calculations quickly. The Louvain community detection algorithm typically performs well when dealing with large networks and is an unsupervised algorithm that is computationally efficient. Therefore, this algorithm was chosen for this stage. The Louvain algorithm [15], a modularity-based community discovery algorithm, is used to detect non-overlapping community structures in graphs. The goal of optimization is to maximize the modularity of the entire structure of the graph (community network). Modularity is used to describe the density of communities in the graph structure. The algorithm initially treats each node as a community, and considers the neighboring nodes of each community separately, attempting to merge them. It finds the partition method with the largest change in modularity and divides it into the same community. Iterative calculations are performed until there is no further change. The modularity formula is given in Equation (11).

$$Q = \frac{1}{2R} \sum_{uv} \left[w_{uv} - \frac{k_u k_v}{2R} \right] \delta(u,v) \tag{11}$$

where u and v are two nodes in G', with $u \in [1,M]$ and $v \in [1,M]$. R represents the total number of edges in G' and w_{uv} represents the weight of the edge between nodes u and v. k_u and k_v are the degree of nodes u and v, respectively. If nodes u and v are in the same community, $\delta(u,v) = 1$; otherwise, $\delta(u,v) = 0$.

The algorithm has a low time complexity and is suitable for large-scale networks and works well. It is currently one of the most commonly used and efficient algorithms. Therefore, in this paper, this algorithm is used to cluster the deep semantic high-quality phrase graph G' and initially generate the key event family C', as shown in Equation (12).

$$C = \text{Louvain}\big(G'\big) \tag{12}$$

where C is the set of clusters generated using Louvain's algorithm, $C = \{C_1, C_2, \ldots, C_r\}$, and r is the total number of communities found. Clusters with node number 1 are omitted, leaving each cluster containing two or more nodes, denoted as C'.

Extraction of Key Event Document: Based on Equation (12), the news documents in corpus D are divided into sets (or clusters) of key event documents corresponding to each cluster of deep sub-semantic high-quality phrases in C'. However, using only phrases for matching may lose documents that express the same event but with different wording. The article selection module IKEDS [7], based on the results of community discovery, enables the selection of documents for the same event but using different wording. This selection module, IKEDS, is used in this paper to further process the community discovery result C' of the deep high-quality phrase graph in order to extract the complete (identically and differently worded) key event documents corresponding to each cluster of deep semantic high-quality phrases in C', denoted by K_i, which represents the set of all documents for the ith key event extracted. Based on the above method, Equation (13) is constructed to extract the complete set of key event documents corresponding to C', denoted by $\mathcal{K}$.

$$\mathcal{K} = \text{IKEDS}(C') \tag{13}$$

where $\mathcal{K} = \{K_1, K_2, \ldots, K_l\}$, $K_i \subset D$. l represents the number of key events.

3.4. Generation of Key Events Context

News events are important events in society that attract attention, so there is a surge in news coverage after a news event has occurred. Simply reading the news is inefficient for objectives such as opinion analysis, which aims to understand the context and trends of news events. Therefore, it is important to generate a news context based on the news text of the key event and an estimate of the segment in which it occurred, after the key event has been detected. This involves two tasks: generating a summary description of each key event and obtaining an estimate of the time span in which each key event occurred.

3.4.1. Abstract Generation

The key event detection task aims to detect key events from the corpus on the same topic, i.e., a set of thematically coherent documents on a specific topic. However, for these key events, it is not possible to visualize the specific description of the key events from the text. For this reason, in this paper, the top-k articles from the document clusters under each key event are selected, and the descriptions of the key events are automatically generated based on a multi-text summary generation method.

In this paper, we use the unsupervised extractive summary generation method Text-Rank [25]. This method does not require the construction of a summary tag for the text, is simple to deploy, and works well without the limitation of the length of the input text. Therefore, this method is used for the generation of summaries. The method first extracts key sentences, then constructs a graph of similarity between sentences, and then calculates the ranking of the sentences according to the similarity. The formula used in this paper to calculate the similarity between two sentences is as follows:

$$Similarity(S_i, S_j) = CosSim\left(\frac{\sum_{k=1}^{count(S_i)} w_k^i}{count(S_i)}, \frac{\sum_{k=1}^{count(S_j)} w_k^j}{count(S_j)}\right) \tag{14}$$

where S_i, S_j are any two sentences in the text, $CosSim(.)$ is the cosine similarity calculation function, $count(.)$ is used to calculate the number of words in the sentence, and w_k^i is the word vector of the kth word in the ith sentence. The average of the word vectors of all the words in the sentence is used as the representation of the vector of the sentence.

After using the Text-Rank algorithm, the sentences in a multi-document document are sorted and the top-n sentences are finally selected as summary. Let l key events $\mathcal{K} = \{K_1, K_2, \ldots, K_l\}$ be predicted, and the top-k news articles in each key event K_i are used to generate summary S_i as the description of key event K_i using Text-Rank.

3.4.2. Identifying Event Context

When identifying the event context, in addition to the textual descriptions of each key event, the generation of a timeline for each key event is also required. The events are arranged in chronological order so that the sequence of events can be determined. In the section above, we have obtained a description of each key event and the description of that event. Therefore, here it is necessary to obtain an estimate of the time of occurrence of each key event from the corpus.

In the news corpus, each news item contains its publication time t. In a key event K_i, there is a series of related news articles in which there is an earliest news release time t_{start}, and a latest news release time t_{end}. Thus, the time span $T_i = \{t_{start}\ , t_{end}\}$ of a key event can be estimated based on the publication time of each article in a key event. The pseudocode of the process is shown as Algorithm 1.

Algorithm 1 Extract Time Estimation

```
Input: Key Event $K$
Output: Time Estimation $T$
t_start ← MAX_TIME
t_end ← MIN_TIME
for d ← $K$ do
    publish_time ← d.publish_time
    if publish_time < t_start do
        t_start ← publish_time
    end if
    if publish_time > t_end do
        t_end ← publish_time
    end if
end for
$T$ ←(t_start, t_end)
return $T$
```

Based on the summary S_i of each key event document cluster and the time estimate T_i, the event context context SL can be obtained, as shown in the following equation:

$$SL = \bigcup_{i=1}^{l} (\{S_i\} + T_i) \tag{15}$$

where $SL = \{sl, sl_2, \ldots, sl_l\}$, $SL_i = \{T_i, S_i\}$, where l is the number of predicted key events. Ranking SL according to the start time t_{start} in the time span estimate T_i, the final event context pulse result SL' is obtained.

4. Experiments

In this section, selected baseline methods for key event detection are compared with the method proposed in this paper in terms of metrics such as precision, recall, and f-measure. Ablation experiments were conducted to verify the effectiveness of the deep semantic information of the nodes in the graph extracted by the GCN proposed in this paper. It also identifies the key event context.

4.1. Experiment Setup

This experiment was conducted on a machine with 128 GB memory, a CPU model of Intel(R) Xeon(R) CPU E5-2620 v4, and a GPU model of NVIDIA GeForce GTX 1080 Ti. The operating system uses Ubuntu 9.4.0, the programming language uses Python 3.7, the deep learning platform uses PyTorch, and PyCharm is used as the integrated context environment.

Experiments were conducted using the Ebola dataset [12], which includes a large number of news articles on the 'ebola outbreak 2014' published from 18 September 2014 to 31 October 2014, each with its time of publication. It contains 741 news articles and 17 key events.

4.2. Metrics

Three evaluation metrics commonly used in previous key event detection studies [7] were used: precision, recall, and f-measure. Precision was used to assess the independence of the events detected by each cluster. Recall was used to assess the detected completeness of the key events. F-measure is the harmonic mean of precision and recall.

Specifically, let the predicted l document clusters be $\mathcal{K} = \{K_1, K_2, \ldots, K_l\}$; the subsequent calculation needs to consider the top-5 articles in the document cluster K_i, so the number of articles in each cluster needs to be >5, where each K_i is a set of articles ranked according to the key event matching degree, and the higher the ranking indicates the higher the relevance to the key event. Assuming that there are N real key events in D, the existence of N real key events is denoted as $G = \{G_1, G_2, \ldots, G_N\}$. If more than half of the top-five articles in K_j belong to G_i, then K_j and G_i are considered to match each other, noted as $Match(G_i, K_j) = 1$, and $Match(G_i, K_j) = 0$ if there is no match. The equations for calculating precision and recall are defined below.

$$check(G_i)\begin{cases} 1,\ if\ \exists\ C_i\ let\ Match(G_i, K_j) = 1 \\ 0,\ else \end{cases} \tag{16}$$

$$precision = \frac{\sum_{G_i} check(K_i)}{l} \tag{17}$$

$$recall = \frac{\sum_{G_i} check(K_i)}{N} \tag{18}$$

F-measure is the harmonic mean of precision and recall.

4.3. Experimental Results and Analysis

The following key event detection methods were selected for comparison:

- Miranda et al. [26] can classify emerging documents into existing document clusters by training an SVM classifier.
- newsLens [27] clusters documents by processing several overlapped time windows.
- Staykovski et al. [28], which is a modification of newsLens.
- S-BERT [29] uses Sentence Transformers to obtain a vector representation of documents, which is then clustered by processing a time window.
- EvMine [7] clusters documents using graphs constructed from detected bursts of temporal peak phrases within a certain time range.

To avoid contingency, the results of each method were obtained by averaging the results of five experiments, which are shown in Table 1.

Table 1. Key event detection results.

	Prec	**Rec**	**Fmeas**
Miranda et al. [26]	0.444	0.706	0.615
newsLens [27]	0.426	0.824	0.561
Staykovski et al. [28]	0.414	0.706	0.522
S-BERT [29]	0.508	**0.836**	0.631
EvMine [7]	0.829	0.682	0.748
GCS	**0.986**	0.706	**0.824**

As can be seen from Table 1, the proposed method outperforms the other benchmark methods in terms of precision and F-measure on the Ebola dataset. The proposed method outperforms the other benchmark methods by 6 to 13 percentage points in these two metrics. The high independence of the events detected by the model in each cluster can be seen in the precision. The score of F-measure demonstrates the good ability of our method to handle noisy data. Some of the baseline methods produce high recall values, which, after analysis, is due to the fact that these methods produce small but many key events and, therefore, coverage of key events can be high. However, since they also generate many of the same key events, their precision values are lower.

4.4. Ablation Study

In this paper, we primarily propose methods for representing and utilizing time information, as well as extracting deep semantic information using a GCN. To verify the effectiveness of these two methods proposed in this paper, we designed and conducted two different ablation experiments.

The first module we chose to verify is the module for representing and utilizing time information. After removing the GCN updating module, the primary difference between our proposed method and Zhang et al.'s EvMine [12] method is the approach for expressing and utilizing time information. EvMine utilizes temporal information to filter phrases and incorporate time information into the method. In contrast, our proposed method uses the concept of position to represent temporal information. As shown in Table 2, the comparative results are as follows. From the table, it can be seen that our proposed method, after removing the GCN module, outperforms EvMine in both precision and F-measure. This demonstrates the effectiveness of our proposed method for representing and utilizing time information.

Table 2. Comparison of the way time information is processed.

	Prec	**Rec**	**Fmeas**
EvMine [7]	0.829	0.682	0.748
GCS-NoGCN	**0.910**	0.647	**0.758**

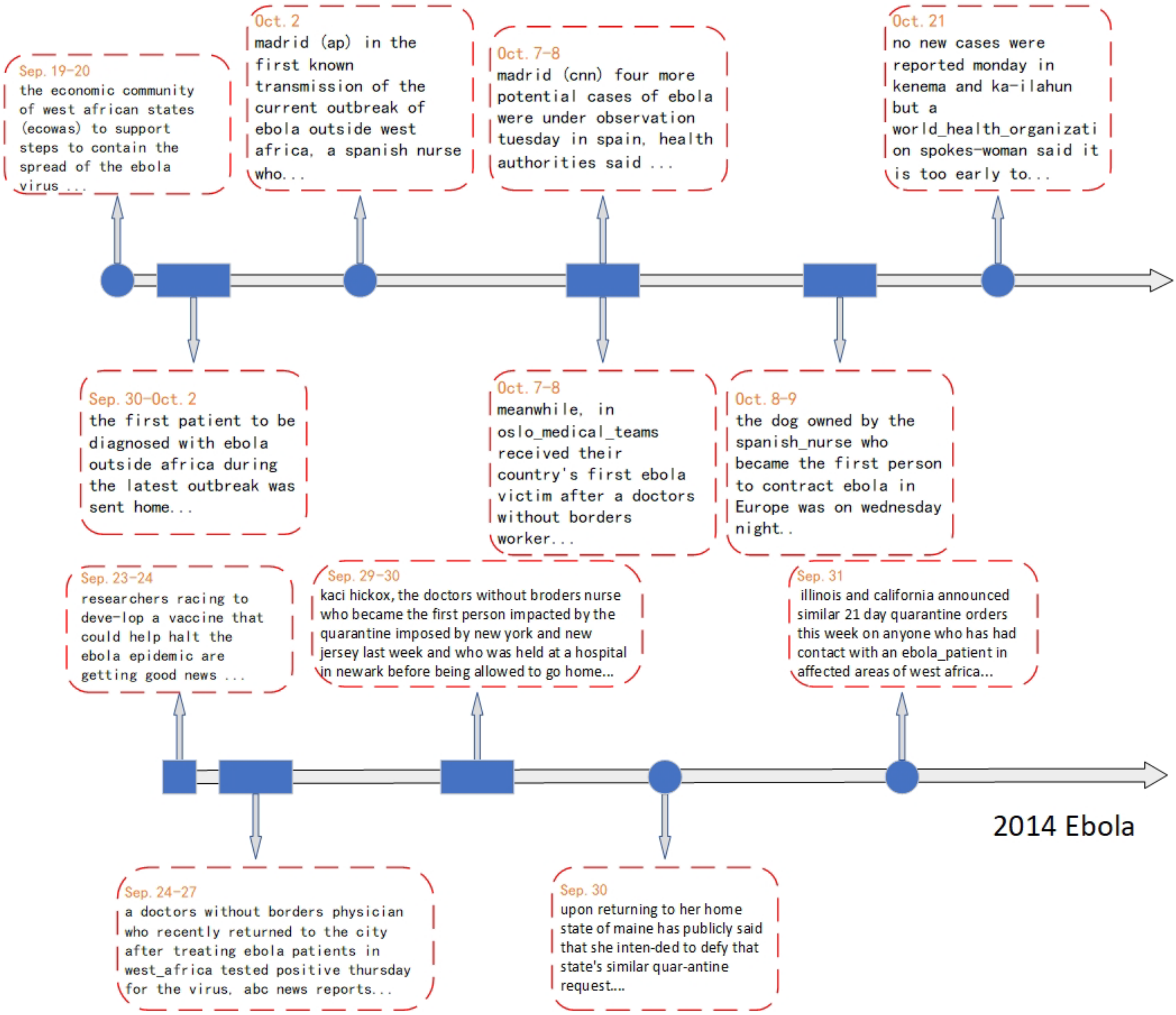
The second module we chose to verify is the module for utilizing the GCN to extract deep semantic information. We compared the performance with and without the GCN module. When removing the GCN module, only the semantic features of the original phrases were used for clustering without utilizing the deep semantic information extracted by the GCN combined with graph structure. The comparative results are shown in Table 3. The results show that the accuracy, recall, and F-measure are significantly improved when the GCN module is added. This fully demonstrates the effectiveness of the proposed module for utilizing the GCN to extract deep semantic information.

Table 3. Proving the validity of the GCN module.

	Prec	**Rec**	**Fmeas**
GCS-NoGCN	0.910	0.647	0.748
GCS	**0.986**	**0.706**	**0.824**

4.5. Identifying the Event Context

As the detected key events are a set of topic-coherent documents on a specific topic, it is not possible to literally visualize the description of a specific key event. Therefore, an unsupervised extractive summary generation algorithm is used to extract a summary S_i of the top-k articles for each key event to obtain a textual description of the key event. This number of articles k can be freely formulated. Here, $k = 10$. S_i is taken to be combined with the time estimate T_i obtained for each key event in Section 3.4. Finally, the event context $SL = \{SL_1, SL_2, \ldots, SL_l\}$, where l is the number of predicted key events and $SL_i = \{T_i, S_i\}$. For the Ebola dataset, the generated event context is shown in Figure 3.



Sep. 19-20
the economic community of west african states (ecowas) to support steps to contain the spread of the ebola virus ...

Oct. 2
madrid (ap) in the first known transmission of the current outbreak of ebola outside west africa, a spanish nurse who...

Oct. 7-8
madrid (cnn) four more potential cases of ebola were under observation tuesday in spain, health authorities said ...

Oct. 21
no new cases were reported monday in kenema and ka-ilahun but a world_health_organization spokes-woman said it is too early to...

Sep. 30-Oct. 2
the first patient to be diagnosed with ebola outside africa during the latest outbreak was sent home...

Oct. 7-8
meanwhile, in oslo_medical_teams received their country's first ebola victim after a doctors without borders worker...

Oct. 8-9
the dog owned by the spanish_nurse who became the first person to contract ebola in Europe was on wednesday night..

Sep. 23-24
researchers racing to deve-lop a vaccine that could help halt the ebola epidemic are getting good news ...

Sep. 29-30
kaci hickox, the doctors without broders nurse who became the first person impacted by the quarantine imposed by new york and new jersey last week and who was held at a hospital in newark before being allowed to go home...

Sep. 31
illinois and california announced similar 21 day quarantine orders this week on anyone who has had contact with an ebola_patient in affected areas of west africa...

Sep. 24-27
a doctors without borders physician who recently returned to the city after treating ebola patients in west_africa tested positive thursday for the virus, abc news reports...

Sep. 30
upon returning to her home state of maine has publicly said that she inten-ded to defy that state's similar quar-antine request....

Figure 3. The result of the key event context.

4.6. Parameter Study

In this section, the impact of the parameter γ on the performance of our proposed method is analyzed. γ is the number of layers of the GCN. Since graph convolutional neural networks can be stacked with layers in the model, the difference in the number of layers will inevitably also affect the performance of the method, so experiments were conducted on the GCN with different numbers of layers. Different numbers of layers from one to five were considered. To avoid chance, all five experiments were performed and averaged, as shown in Figure 4.

We observed that in the initial stages, the method's performance improved as the number of GCN layers increased. This is because the semantic information extracted becomes more enriched with an increasing number of layers. The method achieved its best performance with two to three GCN layers. However, beyond three layers, the method's performance started to degrade. This is due to the information becoming overly smoothed, which leads to a loss of semantic expression ability. Therefore, excessive stacking of layers should be avoided, and setting the parameter to two or three is optimal.

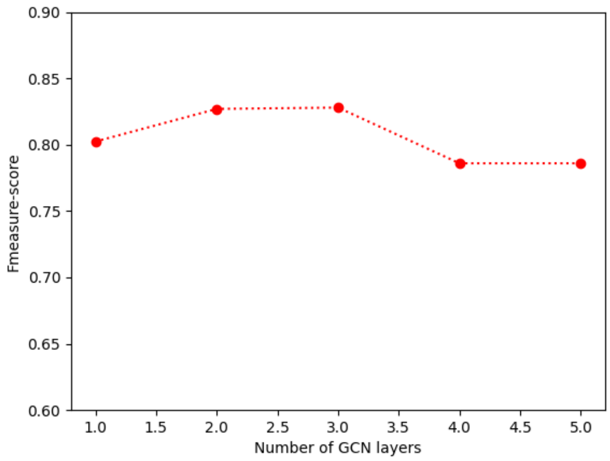


Figure 4. Impact of parameter γ.

4.7. Discussion

In this section, we discuss the advances and limitations of the proposed methodology.

The first aspect we would like to discuss is the advance of our proposed method, which showed significant improvements in accuracy and F-measure values compared to other baseline methods. These results are further supported by ablative experiments, which validate the effectiveness of time-related features, as well as the proposed GCN update module.

Moving on to the limitations of our proposed method, we should note that due to its increased computational complexity, our method takes about 14% longer to process the same dataset compared to similar methods when tested on our experimental platform. It is worth noting, however, that our proposed method achieved significant improvements in evaluation metrics such as accuracy.

5. Conclusions and Future Work

In this paper, a key event detection and event context pulse grooming method is proposed. A method is proposed for the embedding feature representation of the temporal relative position. A temporal non-linear function is also constructed to describe the effect of time on the degree of association between nodes. The subjective use of temporal information in previous studies is avoided. The nodes of the graph are updated using a graph convolutional neural network to extract deep semantic information about individual nodes of a high-quality phrasal graph, thereby enhancing the graph's expressive ability. The summary generation algorithm is applied to each subset of key event news data, which is then sorted according to time span to obtain the final event context, thus avoiding the need to manually generate the relevant descriptions of individual key events. Experiments on the Ebola corpus demonstrate the effectiveness of our proposed approach, achieving higher F-measure and precision values compared to existing baselines methods. There is a little research on this task and the proposed method offers some new ideas and practical implications for current key event detection tasks. For example, it can be used to detect key events in the fields of disease epidemics, natural disasters, etc.

Considering that the use of the original GCN alone may not be sufficient to extract the deep semantic features of the nodes in the graph, future work will try to use more complex graph networks for extraction. This paper currently only performs key event detection on news documents under the same top-level topic, and future work will continue to explore the task of simultaneous key event detection from a corpus containing events from multiple top-level topics.

Author Contributions: Z.L. proposed overarching research goals and aims. Y.Z. and Y.L. performed the experiment. Y.Z. and Y.L. wrote the main manuscript text. C. proposed the research framework. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1406702-3, in part by the National Natural Science Foundation of China under Grant 62006257.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used for this work is the Ebola News dataset published by Zhang [7]. It can be found at https://github.com/yzhan238/EvMine/tree/main/data (accessed on 1 January 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussain, T.; Muhammad, K.; Ullah, A.; Cao, Z.; Baik, S.W.; de Albuquerque, V.H.C. Cloud-assisted multiview video summarization using CNN and bidirectional LSTM. *IEEE Trans. Ind. Inform.* **2019**, *16*, 77–86. [CrossRef]
2. Minaee, S.; Liang, X.; Yan, S. Modern Augmented Reality: Applications, Trends, and Future Directions. *arXiv* **2022**, arXiv:2202.09450.
3. Goyal, P.; Kaushik, P.; Gupta, P.; Vashisth, D.; Agarwal, S.; Goyal, N. Multilevel Event Detection, Storyline Generation, and Summarization for Tweet Streams. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 8–23. [CrossRef]
4. Salwen, Z.C.; Michael, B.; Garrison, B.; Driscoll, P.D. (Eds.) Communication Research Trends. In *Online News and the Public*; Routledge: London, UK, 2006; Volume 25, pp. 37–39.
5. Yang, S.; Sun, Q.; Zhou, H.; Gong, Z.; Zhou, Y.; Huang, J. A topic detection method based on Key Graph and community partition. In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Shenzhen, China, 8–10 December 2018; ACM: New York, NY, USA, 2018; pp. 30–34.
6. Ge, T.; Pei, W.; Ji, H.; Li, S.; Chang, B.; Sui, Z. Bring you to the past: Automatic Generation of Topically Relevant Event Chronicles. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 575–585.
7. Zhang, Y.; Guo, F.; Shen, J.; Han, J. Unsupervised Key Event Detection from Massive Text Corpora. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22), Washington, DC, USA, 14–18 August 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 2535–2544. [CrossRef]
8. Xie, J.; Sun, H.; Zhou, J.; Qu, W.; Dai, X. Event Detection as Graph Parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP, Online*; Association for Computational Linguistics: Toronto, ON, Canada, 2021; pp. 1630–1640. [CrossRef]
9. He, L.; Meng, Q.; Zhang, Q.; Duan, J.; Wang, H. Event Detection Using a Self-Constructed Dependency and Graph Convolution Network. *Appl. Sci.* **2023**, *13*, 3919. [CrossRef]
10. Saravanakumar, K.K.; Ballesteros, M.; Chandrasekaran, M.K.; McKeown, K. Event-Driven News Stream Clustering using Entity-Aware Contextual Embeddings. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; Association for Computational Linguistics: Toronto, ON, Canada, 2021; pp. 2330–2340. [CrossRef]
11. Sia, S.; Dalmia, A.; Mielke, S.J. Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too! In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Toronto, ON, Canada, 2020; pp. 1728–1736. [CrossRef]
12. Santos, J.; Mendes, A.; Miranda, S. Simplifying Multilingual News Clustering Through Projection From a Shared Space. *arXiv* **2022**, arXiv:2204.13418. [CrossRef]
13. Gaglio, S.; Re, G.L.; Moranam, M. A framework for real-time Twitter data analysis. *Comput. Commun.* **2016**, *73*, 236–242. [CrossRef]
14. Brochier, R.; Guille, A.; Velcin, J. Inductive document network embedding with topic word attention. In Proceedings of the Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 14–17 April 2020; Proceedings, Part I 42. Springer: New York, NY, USA, 2020; pp. 326–340.
15. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *10*, 10008. [CrossRef]
16. Guille, A.; Favre, C. Event detection, tracking, and visualization in Twitter: A mention-anomaly-based approach. *Soc. Netw. Anal. Min.* **2015**, *5*, 18–32. [CrossRef]
17. Chen, Z.; Mukherjee, A.; Liu, B.; Hsu, M.; Castellanos, M.; Ghosh, R. Discovering Coherent Topics Using General Knowledge. In Proceedings of the 22st ACM International Conference on Information and Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; ACM: New York, NY, USA, 2013; pp. 209–218.

18. Guzman, J.; Poblete, B. Online relevant anomaly detection in the Twitter stream an eficient bursty keyword delection mode. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, Chicago, IL, USA, 11 August 2013; ACM: New York, NY, USA, 2013; pp. 31–39.
19. Gu, X.; Wang, Z.; Bi, Z.; Meng, Y.; Liu, L.; Han, J.; Shang, J. Ucphrase: Unsupervised context-aware quality phrase tagging. *arXiv* **2021**, arXiv:2105.14078. [CrossRef]
20. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep sentence embedding using long short termmemory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 696–707. [CrossRef]
21. Zhang, Y.; Liu, Q.; Song, L. Sentence-State LSTM for Text Representation. *arXiv* **2018**, arXiv:1805.02474.
22. Thomas, K.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
23. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
25. Rada, M.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 15 January 2004; pp. 404–411.
26. Miranda, S.; Znotins, A.; Cohen, S.B.; Barzdins, G. Multilingual Clustering of Streaming News. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4535–4544.
27. Laban, P.; Hearst, M.A. newsLens: Building and visualizing long-ranging news stories. In Proceedings of the Events and Stories in the News Workshop, Vancouver, BC, Canada, 4 August 2017; pp. 1–9.
28. Staykovski, T.; Barrón-Cedeño, A.; Da San Martino, G.; Nakov, P. Dense vs. Sparse Representations for News Stream Clustering. In *Text2Story@ ECIR 2019*; ACM: New York, NY, USA, 2019; pp. 47–52.
29. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3982–3992.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.