Effective Hierarchical Information Threading using Network Community Detection

Hitarth Narvala, Graham McDonald, and Iadh Ounis

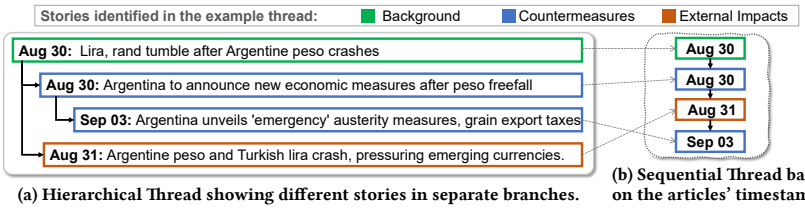
University of Glasgow, Glasgow, UK
h.narvala.1@research.gla.ac.uk
{graham.mcdonald,iadh.ounis}@glasgow.ac.uk

Abstract. With the tremendous growth in the volume of information produced online every day (e.g. news articles), there is a need for automatic methods to identify related information about events as the events evolve over time (i.e., information threads). In this work, we propose a novel unsupervised approach, called *HINT*, which identifies coherent **H**ierarchical **In**formation **T**hreads. These threads can enable users to easily interpret a hierarchical association of diverse evolving information about an event or discussion. In particular, HINT deploys a scalable architecture based on network community detection to effectively identify hierarchical links between documents based on their chronological relatedness and answers to the 5W1H questions (i.e., who, what, where, when, why & how). On the NewSHead collection, we show that HINT markedly outperforms existing state-of-the-art approaches in terms of the quality of the identified threads. We also conducted a user study that shows that our proposed network-based hierarchical threads are significantly ($p < 0.05$) preferred by users compared to cluster-based sequential threads.

1 Introduction

In the digital age, the rise of online platforms such as news portals have led to a tremendous growth in the amount of information that is produced every day. The volume of such information can make it difficult for the users of online platforms to quickly find related and evolving information about an event, activity or discussion. However, presenting this information to the users as a hierarchical list of articles, where each branch of the hierarchy contains a chronologically evolving sequence of articles that describe a story relating to the event, would enable the users to easily interpret large amounts of information about an event's evolution. For example, Figure 1(a) presents different stories that are related to the event "Lira, rand and peso crash" as separate branches of a hierarchical list. We refer to this structure of information as a *Hierarchical Information Thread*. Figure 1(a) illustrates the following three characteristics of hierarchical threads: (1) all of the articles in the thread present coherent information that relates to the same event, (2) different stories (i.e., branches) capture diverse information relating to the event, and (3) the articles that discuss a story are chronologically ordered.

Compared to hierarchical threads, a sequential thread cannot simultaneously capture both the chronology and the logical division of diverse information about



(a) Hierarchical Thread showing different stories in separate branches. **(b) Sequential Thread based on the articles' timestamp.**

Fig. 1: Comparative example of Hierarchical & Sequential Information Threads.

an event. For example in Figure 1(b), a simple chronological order of the articles cannot represent the articles about "Countermeasures" as a coherent story in the thread. In contrast, hierarchical threads (Figure 1(a)) can enable the users to find diverse stories about the event's evolution in an easily interpretable structure.

We propose a novel unsupervised approach, *HINT*,[1] for identifying **H**ierarchical **In**formation **T**hreads by analysing the network of related articles in a collection. In particular, we leverage article timestamps and the 5W1H questions (Who, What, Where, When, Why and How) [8] to identify related articles about an event or discussion. We then construct a network representation of the articles, and identify threads as strongly connected hierarchical network communities.

We evaluate the effectiveness of HINT on the NewSHead collection [7], in both an offline setting and a user study. In our offline evaluation, we show that HINT markedly improves the quality of the threads in terms of Normalised Mutual Information (NMI) and Homogeneity (h) (up to +232.08% NMI & +400.71% h) compared to different established families of related methods in the literature, i.e., document threading [6] and event extraction [12] approaches. We also compare the effectiveness of our hierarchical information threading approach with a recent work on cluster-based sequential information threading [14], which we refer to as SeqINT. In terms of thread quality, we show that HINT is more effective in generating quality threads than SeqINT (+10.08% NMI and +19.26% h). We further conduct a user study to evaluate the effectiveness of HINT's hierarchical threads compared to SeqINT's sequential threads. Our user study shows that the users significantly ($p < 0.05$) preferred the HINT threads in terms of the event's description, interpretability, structure and chronological correctness than the SeqINT threads. We also analyse the scalability of HINT's architecture by simulating a chronologically incremental stream of NewSHead articles. We show that the growth in the execution time of HINT is slower compared to the growth in the number of articles over time.

2 Related Work

Existing tasks such as topic detection and tracking (TDT) [2] and event threading (ET) [13] broadly relate to the problem of identifying information about events. TDT and ET tasks typically focus on identifying *clusters* of events to capture related information about evolving topics or dependent events. However, unlike hierarchical information threads, these clusters of events do not provide a finer-grained view of an event's evolving stories, which makes it difficult for the users to find relevant stories based on their interests.

[1] HINT's code is available at: `https://github.com/hitt08/HINT`

Topic Detection and Tracking (TDT) [2] is the task of identifying threads of documents that discuss a topic, i.e., *topic-based* threads about the chronological evolution of a topic. TDT approaches (e.g. [1,5,18,24]) focus on identifying topical relationships between the documents to automatically detect topical clusters (i.e., topics), and to track follow-up documents that are related to such topics. These topics are often a group of many related events [24]. Differently from topic-based threads about many related events, hierarchical information threads describe evolving information about different *stories* that relate to a *specific* event.

Event threading approaches (e.g., [12,13,20]) first extract events as clusters of related documents, and then identify threads of the event clusters. Differently from event threading, our focus is to identify hierarchical information threads of documents that describe different stories about a single event, activity or discussion. We used the EventX [12] event extraction approach as a baseline in our experiments, since it also identifies related documents about specific events.

Another related task is to identify a few *specific* document threads in a collection such as threads about the most important events [6] or threads that connect any two given documents in the collection [19]. Our work on hierarchical information threading is different in multiple aspects from the aforementioned document threading approaches that aim to identify specific threads in a collection. First, we focus on identifying threads about all of the events in a collection. Second, unlike document threading approaches that use document term features, we focus on the 5W1H questions and chronological relationships between documents to identify evolving information about events. Lastly, unlike existing document threading approaches that generate sequential threads, we propose hierarchical threading to describe various aspects about an event (e.g. different stories).

Recently, Narvala et al. [14] introduced an information threading approach. They deploy clustering to identify *sequential* threads using 5W1H questions and the documents' timestamps (we refer to this approach as SeqINT). Unlike the cluster-based SeqINT approach, in this work, we focus on identifying threads of *hierarchically* associated documents using network community detection methods to capture the evolving stories of an event. Moreover, the SeqINT approach only supports static collections, whereas, our proposed network-based approach can also be deployed to generate information threads in dynamic collections.

3 Proposed Approach: HINT

In this section, we present our proposed approach for identifying Hierarchical Information Threads (HINT). Our approach leverages the chronological relationships between documents, 5W1H questions' answers along with the entities that are mentioned in multiple documents in a collection, to define a directed graph structure of the collection (i.e., a network of documents). We then deploy a community detection algorithm to identify coherent threads by identifying hierarchical links in the network of documents. Figure 2 shows the components of HINT, which we describe in this section, i.e., (1) 5W1H Extraction, (2) Constructing a Document-Entity Graph, (3) Constructing a Directed Graph of the Documents, (4) Nearest Parent Community Detection, and (5) Candidate Thread Selection.

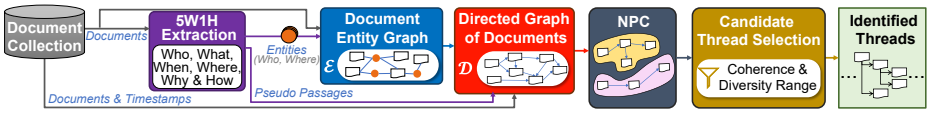


Fig. 2: Components of HINT

5W1H Extraction: We first extract the phrases of text that answers the 5W1H questions from each document in the collection using the Giveme5W1H approach [8]. We then concatenate all of the 5W1H questions' answers for each document as a pseudo-passage (i.e., one pseudo-passage per document). To vectorise the pseudo-passages, we use transformer-based [23] contextual embeddings to capture the context of the events described by the pseudo-passages. We use these embeddings when constructing the Directed Graph of Documents.

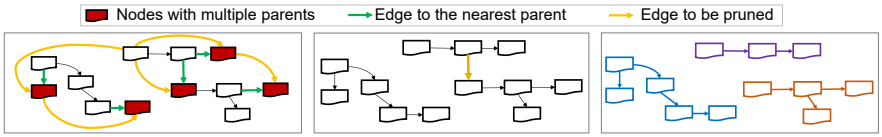
Constructing Document-Entity Graph: After 5W1H extraction, we construct an undirected document-entity graph, $\mathcal{E}$, to identify the common entities between the documents in the collection. The graph $\mathcal{E}$ comprises two types of nodes, i.e., the entities and documents in the collection. We first identify the key entities associated with an event by leveraging the 5W1H questions' answers. In particular, we re-use the available answers to the "who" and "where" questions, which directly correspond to named-entities, i.e., "person/organisation" (who) and "place" (where). In other words, we re-purpose the available named-entity information from the 5W1H extraction to avoid needing an additional named-entity recogniser. We then create an edge between the documents and their corresponding entities, i.e., at most two edges per document node (who and/or where).

Constructing a Directed Graph of Documents: We then use the 5W1H questions' answers, the document-entity graph $\mathcal{E}$ along with the creation timestamps of the documents to construct a document graph, $\mathcal{D}$, from which we identify candidate hierarchical threads. In the graph $\mathcal{D}$, the nodes are the documents in the collections. We define directed edges between documents in $\mathcal{D}$ based on the document timestamp such that the edges between two documents go forward in time. In addition, we define weights for the edges based on the relatedness of the child node to the parent node in a directed edge between two documents. In particular, to effectively capture the relatedness of nodes based on the event they describe, the weight of each edge is defined as: (1) the similarity between the 5W1H pseudo-passages of the documents, (2) the chronological relationship between the documents, and (3) the number of entities mentioned in both of the documents.

To calculate the edge weights of the graph, we first compute the cosine similarity ($cos(p_x, p_y)$) of the 5W1H pseudo-passage embeddings, p_x & p_y (for documents x & y respectively). To capture the chronological relationship between x & y, we compute the documents' time-decay (inspired by Nallapati et al. [13]), i.e., the normalised time difference between the creation times of x & y, defined as:

$$td(x, y) = e^{-\alpha \frac{|t_x - t_y|}{T}} \quad (1)$$

where t_x & t_y are the creation timestamps of documents x & y respectively, T is the time difference between the oldest and latest document in the collection and α is a parameter to factor in time decay. In a dynamic collection, the value of T can be dynamically estimated based on the maximum time difference between articles in the existing threads identified from the historical articles.



(a) For nodes with multiple parents keep exactly one parent with the shortest edge. (b) Identify and prune significantly longer edges in a community. (c) Output connected components as hierarchical threads.

Fig. 3: Nearest Parent Community Detection

We then use the document-entity graph $\mathcal{E}$ to calculate an entity similarity score for each pair of documents in the graph $\mathcal{D}$. To compute an entity similarity score for a pair of documents, x & y, we first identify the number of paths ($|\mathbb{P}_{xy}|$) that connect x & y in the graph $\mathcal{E}$ through exactly one entity node. Second, if $|\mathbb{P}_{xy}| = 0$, we identify the length of the shortest path ($|s_{xy}|$) that connects x & y through multiple entities or other document nodes in $\mathcal{E}$. Intuitively, for documents that have common entities, a higher value of $|\mathbb{P}_{xy}|$ denotes a higher similarity between documents x & y, with respect to the entities that are mentioned in the documents. In contrast, for documents that do not have any common entities (i.e., $|\mathbb{P}_{xy}| = 0$), a longer length of the shortest path, $|s_{xy}|$, denotes less similarity between x & y. Based on the aforementioned description of $|\mathbb{P}_{xy}|$ and $|s_{xy}|$, we define the overall entity similarity score between documents x & y as follows:

$$es(x,y) = \frac{\lambda}{2}(1 + (1 - e^{-\gamma \frac{|\mathbb{P}_{xy}|}{M}})) + \frac{(1-\lambda)}{2} e^{-\gamma \frac{|s_{xy}|}{N}}, \quad \lambda = \begin{cases} 1, & \text{if } |\mathbb{P}_{xy}| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where, M is the largest number of common entities between any two documents in the collection, N is the largest shortest path in the collection, and γ is a parameter to control the relative weights of the number of common entities or the length of the shortest path between x & y.

Lastly, we define the edge weights in the document graph $\mathcal{D}$ (i.e., the distance between x & y) using Equations (1) and (2), and the 5W1H cosine similarity, as:

$$w(x,y) = 1 - cos(p_x, p_y) \cdot td(x,y) \cdot es(x,y) \tag{3}$$

Nearest Parent Community Detection (NPC): From the Directed Graph $\mathcal{D}$, we identify hierarchically connected communities for thread generation. We propose a Nearest Parent Community Detection (NPC) method that identifies strongly connected components of graph $\mathcal{D}$ as communities of hierarchically linked documents. The NPC algorithm is presented in Algorithm 1 and is illustrated in Figure 3. To identify hierarchical links between document nodes, as shown in Figure 3(a), NPC first identifies the nodes that have multiple parents, and follows a greedy approach to keep only the edge that corresponds to the nearest parent (i.e., the edge with the lowest weight; shown with green colour in Figure 3(a)). This selection of only the nearest parent node results in various hierarchically connected components of graph $\mathcal{D}$, as shown in Figure 3(b). However, the connected graph components may still have some weakly connected parent and child nodes (i.e., edges with high weights). Therefore, to remove such weak connections, we split the connected graph components by identifying edges that have significantly higher weights based on the outlier detection method [22]. In particular, within a connected graph component, we determine a threshold edge weight. This threshold value corresponds to the outliers in the distribution of the edge weights within a connected graph component defined as follows [22]:

Algorithm 1: Nearest Parent Community Detection (NPC) Algorithm

```
input  : Directed Graph of Documents 𝒟
output: Connected components of 𝒟 as communities
foreach node n ∈ 𝒟 do
    if inDegree(n) > 1 then
        Find the parent p' that is nearest to n
        foreach p ∈ parents(n) do
            if p ≠ p' then
                Remove edge (p → n)
foreach connected component c ∈ 𝒟 do
    Compute outlier weight threshold for c using Equation (4).
    foreach edge e ∈ c do
        if weight(e) > threshold and outDegree(childNode(e)) > 1 then
            Remove e from 𝒟
```

$$threshold = P_3 + 1.5 * (P_3 - P_1) \tag{4}$$

where P_1 and P_3 are respectively the values for the first and third quartiles (i.e. 25 and 75 percentile) of the edge weight distribution, and $(P_3 - P_1)$ is the interquartile range. We compute this threshold for each connected graph component. While pruning the outlier edges, we do not prune edges where the child nodes do not have any outward edges so that the graph does not contain any isolated nodes. Finally, as shown in Figure 3(c), NPC outputs the connected graph components (i.e., strongly connected communities) as candidate hierarchical threads.

Candidate Thread Selection: From the candidate threads identified by NPC, we select the output threads based on thread coherence and diversity of information. Our focus is on selecting a maximum number of threads from the candidates that are coherent and providing diverse information about their respective events. However, popular metrics (e.g. Cv [16]) for directly computing coherence for all threads in a large collection can be computationally expensive. Therefore, following Narvala et al. [14], to efficiently select candidate threads, we define an estimate of coherence and diversity using the following three measures: (1) The number of documents in a thread $\mathbb{T}$ (i.e., the thread length $|\mathbb{T}|$), (2) The time period, $\mathbb{T}_{span}$, between the timestamps of the first and last documents in a candidate thread, and (3) The mean pairwise document cosine similarity, $\mathbb{T}_{MPDCS}$, of a candidate thread, $\mathbb{T}$, calculated over all pairs of consecutive documents in the candidate thread.

Following [14], we optimise a minimum and maximum threshold range of the aforementioned measures based on coherence, diversity and the total number of selected threads using a smaller sample of NewSHead articles. To compute coherence and diversity, we use the C_v metric [16] and KL Divergence [10] respectively.

4 Experimental Setup

We now describe our experimental setup for the offline evaluation where we evaluate the threads quality (Section 5), and for the user study where we evaluate the effectiveness of hierarchical and sequential threads with real users (Section 6).

Dataset: There are very limited datasets available for evaluating information threads. In particular, previous work (e.g. [6,13]) use manually annotated datasets which are not publicly available. Moreover, classical text clustering datasets such as 20 Newsgroups [11] only contain topic labels and not event labels, which are needed to evaluate event-based information threads.

Therefore, we use the publicly available NewSHead [7] test collection, which contains news story labels and URLs to news articles. Each of the NewSHead story label corresponds to a group of 3-5 articles about a story of an event. For our experiments, we crawled 112,794 NewSHead articles that are associated with 95,786 story labels. We combine the articles from multiple stories about an event into a single set, and refer to these sets as the true thread labels. In particular, since the NewSHead stories often share common articles (i.e., overlapping sets), we perform a union of these overlapping story sets, to create the true thread labels. This resulted in 27,681 true thread labels for the NewSHead articles (average of 4.07 articles per thread). In addition, considering the scalability limits of some of the baseline approaches that we evaluate, similar to Gillenwater et al. [6], we split the collection based on the article creation time into three test sets (37,598 articles each). We execute the threading approaches on these test sets separately, and evaluate their effectiveness collectively on all the three test sets.

Baselines: We compare the effectiveness of HINT to the following baselines:
• **k-SDPP** [6]: We first evaluate the k-SDPP document threading approach, using the publicly available implementation of SDPP sampling [9]. Since the length of k-SDPP threads are fixed, we specify k=4, based on the mean length of the NewSHead threads. Moreover, k-SDPP samples a fixed number of threads. We perform 200 k-SDPP runs with sample size 50 from each of the three test sets (i.e., $200 * 50 * 3 = 30,000$ threads, based on 27,681 NewSHead threads).
• **EventX** [12]: Second, we evaluate the EventX event extraction approach, using its publicly available implementation.
• **SeqINT** [14]: Third, we evaluate SeqINT to compare the effectiveness of cluster-based sequential threading with our hierarchical information threading approach (HINT). We use the edge weight function defined in Equation (3) as the distance function for clustering in SeqINT. Unlike HINT, SeqINT requires an estimate of the number of clusters. For our experiments, we use the number of true thread labels in each of the three test sets as the number of clusters in SeqINT.

HINT: We now present HINT's implementation details and configurations.
• **Pseudo-Passage Embedding**: We evaluate two contextual embedding models [15] for representing the 5W1H pseudo-passages namely: *all-miniLM-L6-v2* and *all-distilRoBERTa-v1*. We denote the aforementioned two embedding models as *mLM* and *dRoB*, respectively, when discussing our results in Section 5.1.
• **Community Detection**: We evaluate the effectiveness of NPC compared to two widely-used community detection methods: Louvain [3] and Leiden [21].
• **Parameters**: We tune HINT's parameters based on thread coherence and diversity on a small sample of NewSHead (Section 3), using the following values:
  ◦ $\alpha; \gamma \Rightarrow \{10^i \ \forall \ -3 \leq i \leq 3; \text{step} = 1\}\}$

$\circ\ x \leq |\mathbb{T}| \leq y \Rightarrow \{x, y\} \in \{\{3, i\}\ \forall\ 10 \leq i \leq 100;\ \text{step} = 10\}$,
$\circ\ x \leq \mathbb{T}_{span} \leq y \Rightarrow \{x, y\} \in \{\{0, i\}\ \forall\ 30 \leq i \leq 360;\ \text{step} = 30\}$,
$\circ\ x \leq \mathbb{T}_{MPDCS} \leq y \Rightarrow \{x, y\} \in \{\{0 + i, 1 - i\}\ \forall\ 0 \leq i \leq 0.4;\ \text{step} = 0.1\}$.

5 Offline Evaluation

Our offline evaluation compares the effectiveness of HINT in terms of the quality of generated threads, compared to the baselines discussed in Section 4. We aim to answer the following two research questions:
• **RQ1**: Is HINT more effective for identifying good quality threads than the existing document threading and event extraction approaches?
• **RQ2**: Is our NPC component more effective at identifying communities for thread generation than existing general community detection methods?

Evaluation Metrics: We evaluate thread quality based on the agreement of articles in the generated threads with the NewSHead thread labels. However, we note that thread quality cannot indicate whether the sequence of articles in a thread is correct, which we evaluate later in our user study (Section 6). Intuitively, our offline evaluation considers threads as small clusters of articles. We use the following popular cluster quality metrics to measure the thread quality: Homogeneity Score (h) [17] and Normalised Mutual Information (NMI) [4].

Since all of the NewSHead articles have an associated thread label, we compute h and NMI using all of the articles in the collection to measure the thread quality. Moreover, for each of the evaluated approaches, it is possible that the approach will not include all of the NewSHead articles in the generated threads. Therefore, we also report the number of generated threads along with the total and mean of the number of articles (mean $|\mathbb{T}|$) in each of the generated threads.

5.1 Results

Table 1 presents the number, length and quality of the generated threads. Firstly addressing **RQ1**, we observe from Table 1 that the NPC configurations for HINT markedly outperform the k-SDPP and EventX approaches from the literature along with the SeqINT approach in terms of h and NMI (e.g. NMI; mLM-NPC: 0.797 vs k-SDPP: 0.190 vs EventX: 0.240 vs SeqINT: 0.724). Even though both HINT and SeqINT use 5W1H questions, HINT's NPC community detection and graph construction using time decay and entity similarity contributes to its higher effectiveness over SeqINT. Moreover, since we measure h and NMI on the entire collection, the number of articles identified as threads (e.g. mLM-NPC: 74.67% articles) is an important factor in HINT's effectiveness compared to existing methods. For example, EventX identified only 16.58% articles as threads, which affects its overall effectiveness. To investigate this, we evaluate EventX and HINT using only the NewSHead articles that are identified as threads (16.58% & 74.67% respectively). Even for this criteria, HINT outperforms EventX (e.g. 0.927 vs 0.883 NMI). We further observe that the number of threads identified are markedly higher for HINT (e.g. mLM-NPC: 18,340) compared to k-SDPP (4,599), EventX (7,149), and SeqINT (13,690). Furthermore, we observe that

Table 1: Results for the Thread Quality of the evaluated approaches. (True #articles=112,794; #threads=27,681 and mean $|\mathbb{T}|$=4.07).

Configuration	h	**NMI**	**#Articles**	**#Threads**	**mean** $\|\mathbb{T}\|$
K-SDPP	0.107	0.190	13,076	4,599	2.84
EventX	0.141	0.240	18,698	7,149	2.62
$\text{SeqINT}_{\text{mLM}}$	0.592	0.724	69,430	13,690	5.07
$\text{SeqINT}_{\text{dRoB}}$	0.541	0.684	63,336	12,522	5.06
$\text{HINT}_{\text{mLM-Louvain}}$	0.001	0.003	207	20	10.35
$\text{HINT}_{\text{dRoB-Louvain}}$	0.001	0.003	202	15	13.47
$\text{HINT}_{\text{mLM-Leiden}}$	0.001	0.001	78	17	**4.59**
$\text{HINT}_{\text{dRoB-Leiden}}$	0.001	0.001	69	14	4.93
$\text{HINT}_{\text{mLM-NPC}}$	**0.706**	**0.797**	**84,228**	**18,340**	**4.59**
$\text{HINT}_{\text{dRoB-NPC}}$	0.686	0.783	81,770	17,819	**4.59**

the mean number of articles per thread (mean $|\mathbb{T}|$) for HINT (4.59) is the closest to the true threads (4.07) in NewSHead. Therefore, for RQ1, we conclude that HINT is indeed effective for generating quality information threads compared to existing document threading (k-SDPP) and event extraction (EventX) approaches as well as cluster-based information threading (SeqINT).

Moving on to **RQ2**, from Table 1 we observe that the Louvain and Leiden configurations of HINT are the least effective. Upon further investigations, we found that these general community detection methods identify comparatively larger communities than NPC, which can affect the coherence of the generated threads. Therefore, the candidate selection component in HINT when using Louvain or Leiden selects a very small number of threads (e.g., mLM-Louvain: 20, mLM-Leiden: 17, compared to mLM-NPC: 18,340). Therefore, in response to RQ2, we conclude that our proposed NPC is the most suitable method to identify the strongly connected communities for effective thread generation.

5.2 Ablation Study

We now present an analysis of the effectiveness of different components of HINT.
• **Effect of Time-Decay and Entity Similarity**: We first analyse the effectiveness of the time-decay and entity similarity scores to compute the weights of the edges in the Document Graph ($\mathcal{D}$). In particular, we evaluate HINT in two additional settings to compute the edge weights: (1) cosine similarity of the 5W1H pseudo-passages (i.e., by setting $td(x, y) = es(x, y) = 1$ in Equation (3)), and (2) cosine similarity and time-decay (TD) (i.e., $es(x, y) = 1$). From Table 2, we observe that our proposed configuration to compute the edge weights with both time-decay and entity similarity (e.g. mLM-TD-ENT: 0.797 NMI) outperforms other configurations that include only cosine similarity (e.g. mLM: 0.759 NMI) or cosine and time-decay similarity (e.g. mLM-TD: 0.796 NMI). However, we also observe that the improvements from including the time-decay similarity are larger compared to including both time-decay and entity similarity. As future work, we plan to investigate whether including an entity recognition component in addition to the 5W1H extraction can further improve the thread quality.

Table 2: Effect of Time-Decay and Entity Similarity on thread quality.

Configuration	h	NMI
mLM	0.657	0.759
dRoB	0.642	0.747
mLM-TD	0.705	0.796
dRoB-TD	0.686	0.783
mLM-TD-ENT	**0.706**	**0.797**
dRoB-TD-ENT	0.686	0.783

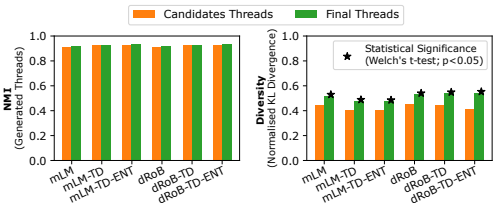


Fig. 4: Effect of Candidate Selection on NMI and Diversity.

• **Effect of Candidate Thread Selection**: We also analyse the effect of the candidate selection on the quality and diversity of the generated threads. We use KL divergence [10] to measure the threads' diversity of information. For a given thread, we hold-out each document from the thread and compute the KL divergence between the probability distributions of the words in the held-out document and the words in the rest of the documents in the thread. Since in this analysis we are focused on the quality of the generated threads, we compute h and NMI using only the articles that are identified as part of the *generated* threads.

Figure 4 shows the thread quality (NMI) and information diversity of the candidate threads identified by the NPC and the final output threads from the candidate selection component. We first observe that the quality of the candidate threads and the final threads are comparable. However, the final threads are significantly (Welch's t-test; $p < 0.05$) more diverse than the candidate threads. Therefore, our proposed candidate selection component can effectively select quality information threads that describe diverse information about an event.

6 User Study

As we described in Section 3, our proposed HINT approach captures hierarchical links between documents. These hierarchical links can present chronological hierarchies and a logical division of diverse information, e.g. different stories that are each related to the same event. However, unlike HINT's hierarchical threads, sequential threads (such as from SeqINT) may not be able to capture such logical division of diverse information. Therefore, it is important to know which of these presentation strategies (i.e., hierarchical or sequential) is preferred by users. We conducted a user study that evaluates whether hierarchical information threads are more descriptive and more interpretable to users than sequential threads. In particular, we compared HINT with the best performing baseline from our offline evaluation, i.e., SeqINT. We selected the best configurations of HINT and SeqINT from our offline evaluation (i.e., $\text{HINT}_{\text{mLM-NPC}}$ & $\text{SeqINT}_{\text{mLM}}$).

Our user study aims to answer the following two research questions:

• **RQ3**: Do users prefer the hierarchical threads that are generated by HINT compared to the cluster-based sequential SeqINT threads?

• **RQ4**: Do the hierarchical links between articles in HINT threads effectively present a logical division of diverse information about an event?

Experiment Design: We follow a within-subject design for our user study, i.e., we perform a pairwise evaluation of the threads generated by the HINT and

SeqINT approaches. In other words, each user in the user study evaluates pairs of threads, where each pair of threads is about the same event, but the threads are generated from different threading approaches (i.e., HINT and SeqINT). When selecting the threads to present to the users, we select the best pairs of threads based on the threads' precision scores calculated over both of the threads in a pair, i.e., the ratio of the number of articles associated with a single true thread label to the total number of articles in a thread. In addition, we select threads that have exactly 4 articles based on the mean number of articles in the NewSHead thread labels. Overall, we selected 16 pairs of threads. We then distributed the selected pairs into 4 unique sets (i.e., 4 pairs per set), such that each of our study participants evaluates the pairs of threads from a particular set.

The user study participants were asked to select their preferred thread from each of the pairs with respect to each of the following criteria: (1) the description of an event in the thread, (2) the interpretability of the thread, (3) the structure of the thread, and (4) the explanation of the event's evolution in the thread. We also asked participants to rate each of the threads with respect to the thread's: (1) coherence, (2) diversity of information, and (3) chronology of the presented articles. We deployed a 4-point likert scale to capture the participants' ratings. The choice of a 4-point scale was based on the number of articles that we fix (i.e. 4) in each of the threads. Additionally, participants were asked to rate the HINT threads with respect to the logical division of the information in the branches of the thread (i.e., the logical hierarchies). The participants were presented with the title of the articles in each of the threads, as illustrated in the example in Figure 1.

Participant Recruitment: We recruited 32 participants using the MTurk (www.mturk.com) crowdsourcing platform. The recruited participants were all 18+ years of age and from countries where English is their first language. From the 32 participants, we first assigned 8 participants to each of the 4 sets of thread pairs. We further created 4 participant groups for each of the sets (i.e., 2 participants per group-set combination), using balanced Latin square counterbalancing by permuting the 4 pairs of threads in each set.

6.1 Results

Figure 5 shows our user study's results in terms of the participants' preferences and ratings. We use the chi-square goodness-of-fit test to measure statistical significance between the participants' preferring the HINT or SeqINT threads, as shown in Table 3. We also use a paired-samples t-test to measure the statistical significance between the participants' ratings for HINT and SeqINT (Table 4).

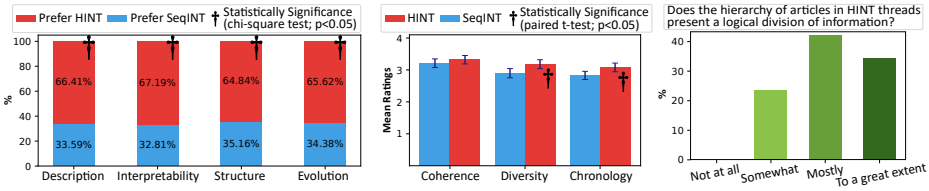
First, addressing **RQ3**, from Figure 5(a) and Table 3, we observe that participants significantly (chi-square test; $p < 0.05$) prefer our proposed HINT approach compared to SeqINT, for all four criteria, i.e. description, interpretability, structure and evolution. Further, from Figure 5(b), we observe that the participants rate the HINT threads higher for all of the three criteria, i.e., coherence, diversity and chronology. Moreover, as shown in Table 4, the participants' ratings for HINT are significantly (t-test; $p < 0.05$) higher with respect to diversity and chronology. However, the improvement in coherence ratings for HINT is not significant compared to SeqINT. This shows that both HINT and SeqINT threads

Table 3: Chi-square goodness-of-fit test results.

Criteria	$\chi^2(1)$	Cohen's w	p	Power
Description	13.781	0.328	< 0.001	96.00%
Interpretability	15.125	0.344	< 0.001	97.33%
Structure	11.281	0.297	0.001	91.93%
Evolution	12.500	0.313	< 0.001	94.30%

Table 4: Paired Samples t-test results.

Criteria	Cohen's d	p	Power
Coherence	0.117	0.187	25.96%
Diversity	0.294	0.001	91.08%
Chronology	0.251	0.005	80.46%



(a) User Preferences (b) User Ratings (c) Logical Hierarchies

Fig. 5: Pairwise participants' preferences and ratings of the threading methods.

can identify related articles about an event. However, HINT threads provide significantly more diverse information about the event, as shown in Figure 5(b). Overall, for RQ3, we conclude that the participants significantly preferred hierarchical HINT threads over the sequential SeqINT threads. Moreover, the participants' ratings show that the HINT threads provide significantly more diverse and chronologically correct information about an event than the SeqINT threads.

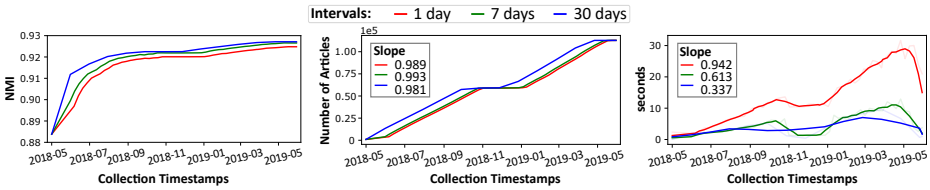
Moving on to **RQ4**, Figure 5(c) shows the participants' ratings for the logical division of information by the different hierarchies in the HINT threads. From Figure 5(c), we observe that the majority of participants (44%) said that the hierarchies in the HINT threads are *mostly* logical. Moreover, none of the participants said that the hierarchies in the HINT threads are *not at all* logical. Therefore, for RQ4, we conclude that HINT threads present a logical presentation of diverse information (i.e. distinct stories) about an event through the hierarchical association between related articles.

Overall, our user study shows that HINT's hierarchical threads are significantly preferred by users compared to sequential threads. Moreover, the study shows that HINT can effectively present a logical hierarchical view of aspects (e.g. stories) about the evolution of an event.

7 Identifying Incremental Threads

We now present an analysis on the scalability of the HINT's architecture. This analysis focuses on the overall efficiency of HINT's novel components, i.e., the document-entity graph ($\mathcal{E}$), document graph ($\mathcal{D}$), NPC, and candidate selection.

We deploy HINT to generate threads incrementally by simulating a chronological stream of NewSHead articles. NewSHead articles were published between May 2018 and May 2019, i.e., over a period of 394 days [7]. We first generate threads from the articles that were published in the first 30 days in the collection (i.e., historical run). From the historical run, we store the NPC communities as a single graph of hierarchically connected document nodes ($\mathcal{D}'$), as illustrated in Figure 3(c). We then simulate three incremental article streams such that,



(a) Cumulative NMI (b) Cumulative Number of Ingested Documents (c) Execution Time (per day)

Fig. 6: Incremental HINT on a simulated stream of NewSHead articles.

in each stream, documents from different sequential time intervals are input to HINT to be added to existing threads or generate new threads, i.e., daily (every 1 day), weekly (every 7 days), and monthly (every 30 days). For each incremental run, we extend the document graph $\mathcal{D}'$ by computing the similarity between the new articles in the stream and the existing articles in $\mathcal{D}'$ using Equation (3). We then perform community detection on $\mathcal{D}'$ using NPC, followed by candidate selection of the newly identified or extended threads.

Figure 6 shows, for each of the incremental streams of the NewSHead articles, the NMI of the generated threads, the total number of ingested documents and HINT's execution time. From Figure 6(a), we observe that the quality (NMI) of the HINT threads quickly increases during the initial 2 months of the incremental runs (i.e., between May and July 2018) and remains comparable thereafter. This shows that HINT is still effective when there are only a small number of articles. Furthermore, Figures 6(b) and 6(c) show that there is a linear increase in the execution time of HINT as the number of ingested articles increases. Most importantly, we observe that the rate of increase in HINT's execution time is slower than the increase in ingested articles (e.g. 0.981 slope as the number of monthly ingested documents increases vs 0.337 slope for the execution time in seconds). Additionally from Figure 6(c), we observe that the rate of increase in the daily execution times is the highest, followed by the weekly and monthly execution times. This suggests that the time taken for incremental executions of HINT can be reduced by increasing the frequency of days between the incremental executions.

Overall, this analysis shows that HINT can effectively and efficiently identify threads in a dynamic collection. Moreover, HINT's architecture is scalable, as the rate of increase in HINT's execution time is slower compared to the increase in the number of ingested articles (Figures 6(b) and 6(c)).

8 Conclusions

We proposed a novel unsupervised approach, HINT, for hierarchical information threading. The hierarchical threads generated by HINT can help users to easily interpret evolving information about stories related to an event, activity or discussion. Our offline evaluation showed that HINT can effectively generate quality information threads compared to approaches from the literature. In addition, our user study showed that HINT's hierarchical information threads are significantly preferred by users compared to cluster-based sequential threads. Moreover, with its scalable network community-based architecture, HINT can efficiently identify threads in a dynamic collection to capture and track evolving information.

References

1. Allan, J.: Topic detection and tracking: event-based information organization, The Information Retrieval Series, vol. 12. Springer US (2012)
2. Allan, J., Carbonell, J.G., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop (1998)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10) (2008)
4. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data Engineering **17**(12) (2005)
5. Fan, W., Guo, Z., Bouguila, N., Hou, W.: Clustering-based online news topic detection and tracking through hierarchical bayesian nonparametric models. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (2021)
6. Gillenwater, J., Kulesza, A., Taskar, B.: Discovering diverse and salient threads in document collections. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2012)
7. Gu, X., Mao, Y., Han, J., Liu, J., Wu, Y., Yu, C., Finnie, D., Yu, H., Zhai, J., Zukoski, N.: Generating representative headlines for news stories. In: Proceedings of The Web Conference (2020)
8. Hamborg, F., Breitinger, C., Gipp, B.: Giveme5W1H: A universal system for extracting main events from news articles. In: Proceedings of the 13th ACM Conference on Recommender Systems, 7th International Workshop on News Recommendation and Analytics (2019)
9. Kulesza, A., Taskar, B.: Structured determinantal point processes. In: Proceedings of the Advances in Neural Information Processing Systems (2010)
10. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics **22**(1) (1951)
11. Lang, K.: NewsWeeder: Learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning (1995)
12. Liu, B., Han, F.X., Niu, D., Kong, L., Lai, K., Xu, Y.: Story Forest: Extracting events and telling stories from breaking news. ACM Transactions on Knowledge Discovery from Data **14**(3) (2020)
13. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event threading within news topics. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management (2004)
14. Narvala, H., McDonald, G., Ounis, I.: Identifying chronological and coherent information threads using 5W1H questions and temporal relationships. Information Processing & Management **60**(3) (2023)
15. 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 (2019)
16. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining (2015)

17. Rosenberg, A., Hirschberg, J.: V-Measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2007)
18. 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 (2021)
19. Shahaf, D., Guestrin, C.: Connecting two (or less) dots: Discovering structure in news articles. ACM Transactions on Knowledge Discovery from Data **5**(4) (2012)
20. Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., Leskovec, J.: Information cartography: Creating zoomable, large-scale maps of information. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2013)
21. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: Guaranteeing well-connected communities. Scientific Reports **9**(5233) (2019)
22. Tukey, J.W., et al.: Exploratory data analysis, vol. 2. Reading, MA (1977)
23. 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 Advances in Neural Information Processing Systems (2017)
24. Zong, C., Xia, R., Zhang, J.: Topic detection and tracking. In: Text Data Mining. Springer Singapore (2021)