Information Processing and Management 60 (2023) 103274

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Identifying chronological and coherent information threads using 5W1H questions and temporal relationships

Hitarth Narvala *, Graham McDonald, Iadh Ounis

University of Glasgow, School of Computing Science, Glasgow, G128QQ, United Kingdom

ARTICLE INFO

Keywords:
Information threading
5W1H
Temporal relationship
Information extraction
Event extraction

ABSTRACT

Due to the massive volume of articles produced online every day, it is challenging for online platforms (e.g., news agencies) to present the information about an event, activity or discussion to their users in an easily digestible format. Therefore, there is a need for automatic methods to extract related and time-ordered information about events (i.e., *information threads*) from large unstructured collections of documents. In this work, we propose a novel unsupervised hierarchical agglomerative clustering (HAC) based information threading approach to generate chronological and coherent threads of information in a collection. Unlike, the well-known tasks of topic detection and tracking or event threading that focus on grouping information by important keywords and/or entities, our proposed approach identifies threads based on temporal relations and diverse information about an event, i.e., who did what, why, where, when and how (aka the 5W1H questions). In particular, our proposed approach, deploys a tailored similarity function for HAC by leveraging extracted answers to 5W1H questions along with time decay between documents. We evaluate our proposed HAC 5W1H information threading approach on two large expert-annotated collections of news articles, i.e., NewSHead and Multi-News (over 112k and 32k articles, respectively). Our experiments show that HAC 5W1H markedly improves the number of, and quality of, threads that are generated compared to existing state-of-the-art approaches from the literature, e.g., 100.98% more threads and +213.39% improvement in Normalised Mutual Information compared to the best evaluated baseline on the larger NewSHead collection. We also conducted a user study that shows that our proposed HAC 5W1H information threading approach is significantly ($p < 0.05$) preferred by users in terms of coherence, diversity and chronological correctness compared to the existing state-of-the-art approaches.

1. Introduction

The volume of articles that are published online every day can make it infeasible for users of search engines to find and make sense of the large amounts of information that can be related to an event, activity or discussion that they are interested in. Therefore, being able to automatically extract related and time-ordered information about events (i.e., information threads) from large unstructured collections of documents can assist online platforms (e.g., news agencies) to present the information to their users in an easily digestible format. For example, a time-ordered thread of news articles about a legal trial (as shown in Fig. 1) can help the users to quickly understand the background, progress and verdict of the trial.

* Corresponding author.
E-mail addresses: h.narvala.1@research.gla.ac.uk (H. Narvala), Graham.Mcdonald@glasgow.ac.uk (G. McDonald), Iadh.Ounis@glasgow.ac.uk (I. Ounis).
URLs: https://www.gla.ac.uk/pgrs/hitarthnarvala/ (H. Narvala), http://www.dcs.gla.ac.uk/~graham/ (G. McDonald), http://www.dcs.gla.ac.uk/~ounis/ (I. Ounis).

https://doi.org/10.1016/j.ipm.2023.103274
Received 27 September 2022; Received in revised form 21 November 2022; Accepted 9 January 2023
Available online 18 January 2023

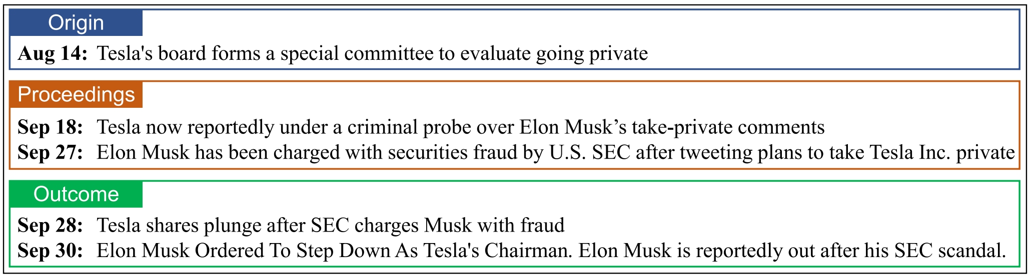


Fig. 1. Example of an information thread describing the origin, proceedings and outcome of a Legal Trial.

Previous work on identifying threads of related information have each presented their own individual use cases and definitions of threads. For example, identifying threads of topically related documents by topic detection and tracking (TDT) (Allan, Carbonell, Doddington, Yamron, & Yang, 1998) approaches or identifying threads of related events by *event* threading (Nallapati, Feng, Peng, & Allan, 2004) approaches. However, in this work, our focus is on a much more generalised concept of threading in a document collection that we refer to as *Information Threads* and define as follows:

Definition 1 (*Information Thread*)**.** A chronological and coherent sequence of documents or passages from multiple documents that capture the *temporal* relationships between documents and describe *diverse* information about a particular event, activity or discussion.

Typically, a coherent thread indicates that the documents associated with the thread describe the same particular event. In addition, temporal relationships between documents can indicate how likely it is that documents that mention the same set of keywords or entities discuss the same event. For example, documents that are published in different time periods are less likely to discuss the same event (Nallapati et al., 2004). Along with the temporal relationships, it is also important to capture diverse information about an event such as *who* was involved in the event, *what* really happened, *where*, *when*, *why* and *how*, i.e., the journalistic 5W1H questions (Hamborg, Breitinger, & Gipp, 2019). In particular, these 5W1H questions together can typically describe either an event, activity or discussion in documents across a collection.

In this work, we propose a novel unsupervised machine learning approach for identifying information threads by leveraging answers to 5W1H questions from documents, the temporal relationships between documents and hierarchical agglomerative clustering (HAC). We first deploy a tailored similarity function that can capture both the temporal relationships between documents and the similarity between the 5W1H questions' answers in the documents. We then perform hierarchical agglomerative clustering of the 5W1H questions' answers from documents that are temporally related to construct time-ordered document sequences as candidate information threads. Finally, we deploy an estimated threshold of coherence and diversity of information to select the candidates as output information threads. In particular, we present a novel architecture that combines the aforementioned processes to effectively identify chronological and coherent information threads in large collections.

We evaluate the effectiveness of our proposed information threading approach, *5W1H-HAC*, in both an offline setting and in a user study. In particular, in the offline setting, we use two large collections of news articles, namely NewSHead (Gu et al., 2020) and Multi-News (Fabbri, Li, She, Li, & Radev, 2019). We also evaluate the quality of threads generated by our proposed 5W1H-HAC approach using the expert-annotated news story labels in NewSHead and Multi-News collections. We compare the quality of the 5W1H-HAC threads with approaches from different established families of related threading methods in the literature: k-SDPP document threading (Gillenwater, Kulesza, & Taskar, 2012) and EventX event extraction (Liu et al., 2020). We also include K-Means (Lloyd, 1982; MacQueen, 1967) clustering as an indicative baseline in our offline evaluation to compare the task of document clustering with information threading. To complement the offline evaluation, we also conduct a user study using the larger NewSHead collection to perform a pair-wise comparison of our proposed 5W1H-HAC approach with the best performing baselines in the offline evaluation (i.e., k-SDPP and EventX) in terms of the coherence, diversity of information and chronological correctness of the generated threads. Our contributions in this paper are 3-fold:

1. We propose a novel approach for identifying chronological and coherent information threads in a collection.
2. We conduct both an offline evaluation and a user study to evaluate the effectiveness and added-value of our proposed 5W1H-HAC information threading approach in terms of the thread quality, coherence, diversity and chronological correctness.
3. We show that the proposed 5W1H-HAC approach outperforms existing *state-of-the-art* approaches in the literature markedly in the offline evaluation and significantly in the user study.

To the best of our knowledge, this is the first work that leverages 5W1H questions to identify chronological and coherent threads of related information in a large document collection. We show that our approach markedly improves the number and the quality of the generated threads consistently on both the NewSHead and Multi-News collections compared to the baselines from the literature. For example, on the NewSHead collection, our 5W1H-HAC approach increases the number of generated threads by up to 100.98%, and improves the quality of the generated threads by up to +213.39% NMI (Normalised Mutual Information) and

+347.28% Homogeneity compared to the best evaluated baseline. Moreover, the participants in our user study rated the threads generated by our proposed approach as significantly (paired samples t-Test, $p < 0.05$) more coherent, diverse and chronologically correct than the existing k-SDPP and EventX approaches.

The remaining sections of the paper are organised as follows. In Section 2 we present related work on identifying threads of related information. We describe our proposed information threading approach, 5W1H-HAC, in Section 3. In Section 4, we present our experimental setup of the offline evaluation and user study, which we discuss in Sections 5 and 6 respectively. We present a detailed analysis of the findings from our experiments in Section 7. In Section 8, we present a discussion of the results and implications of this work. Finally, we provide the conclusions in Section 9.

2. Related work

In this section, we discuss related work on identifying threads of related information in large unstructured collections of documents, i.e., information threads. There have been relatively few approaches proposed in the literature for automatically identifying information threads in document collections. Moreover, previous studies of information threading approaches have each presented a different definition of information threads that is only appropriate for the particular use case that the study has addressed.

2.1. Topic detection and tracking

Topic Detection and Tracking (TDT) (Allan et al., 1998) was one of the early investigations into identifying *topical* threads in news articles. Over the past two decades, TDT has received much attention in the literature (e.g. Allan, 2012; Fan, Guo, Bouguila, & Hou, 2021; Lee, Lee, & Jang, 2007; Mele, Bahrainian, & Crestani, 2019; Saravanakumar, Ballesteros, Chandrasekaran, & McKeown, 2021; Yu, Zhang, Ting, & Sheng, 2007; Zong, Xia, & Zhang, 2021). TDT approaches typically leverage document clustering and/or topic modelling techniques such as K-Means and Latent Dirichlet Allocation (LDA) to detect topics in news articles, and further track the follow-up articles that relate to the discovered topics. As described by Zong et al. (2021), such topics are often referred to as a group of many related events that together form a core event. For example, "Air Strikes in Syria" is a core event that can have many smaller events/activities/discussions, such as the cause of the main event, reactions to this event from different world leaders, and the aftermath of the event. Our work is broadly related to TDT in that we are interested in identifying groups of related documents in a collection. However, differently from document clustering in TDT that is based on identifying topical relationships between documents, in this work we focus on identifying documents that are related at a finer granularity than topics (i.e., documents about specific smaller events instead of a core event topic).

2.2. Document threading

Existing document threading approaches focus on identifying threads between specific documents (Shahaf & Guestrin, 2012) or threads about the most important events in a collection (Gillenwater et al., 2012). Shahaf and Guestrin (2012) presented an approach to connect any given two documents with a coherent sequence of documents. The authors deployed a linear programming based algorithm to determine a thread of a fixed number of documents that connects the specified thread endpoints in a bipartite directed graph of the documents and words in a collection. Gillenwater et al. (2012) presented an approach for identifying a small set of document threads that could describe the most important events in a collection. The authors' approach, named k-SDPP, sampled a set of threads from a graph representation of a document collection, with document similarities represented by weights on the edges of the graph, using a structured determinantal point process (Kulesza & Taskar, 2010). Differently from the work of Shahaf and Guestrin (2012) and Gillenwater et al. (2012) that focused on identifying threads between specific documents or the most salient threads in a collection, we focus on identifying the maximum number of information threads in a collection to help various information seekers to quickly identify relevant information in a time-ordered sequence.

2.3. Event extraction and threading

Another related research direction is to extract events in a collection and further identify threads of the related events. A majority of the existing event extraction approaches (e.g. Aggarwal & Subbian, 2012; Huang et al., 2016; Kuo & Chen, 2007; Qian, Deng, Ye, Ma, & Yuan, 2019; Chen & Wang, 2021; Jacobs & Hoste, 2020) are focused on identifying events as clusters of documents in a collection using document term based features, entities and important keywords. In general, event extraction, has a similarity with our work on identifying clusters of documents that discuss the same event. However, differently from existing event extraction approaches, we focus on 5W1H questions to generate information threads. A major benefit of using 5W1H questions is that, unlike keywords, answers to 5W1H questions can indicate the specific context about an event in which an entity or an important keyword appears in the document. Recently, Zhang, Guo, Shen, and Han (2022) proposed a method for key event extraction to identify documents that are relevant to the key events in a collection. However, extracting only a few documents about the key events is not sufficient for information threading, where the focus is on identifying a maximum number of documents that are associated with threads about events in a large collection. Further to event extraction, event threading approaches (e.g. Liu et al., 2020; Nallapati et al., 2004; Shahaf et al., 2013) leverage the extracted event clusters to identify threads that describe related events. Nallapati et al. (2004) presented one of the initial works on event threading. Differently from TDT approaches that focus on detecting topics,

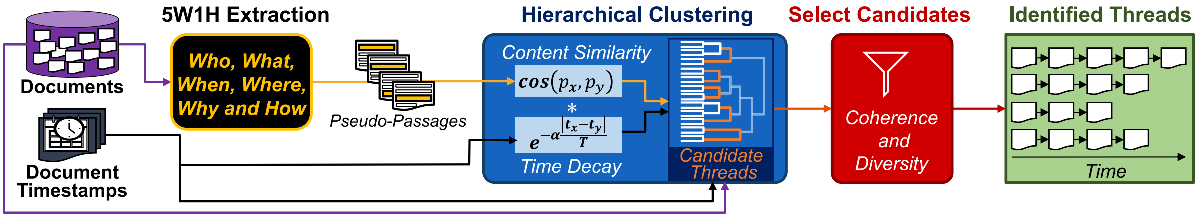


Fig. 2. Components of our 5W1H-HAC information threading approach.

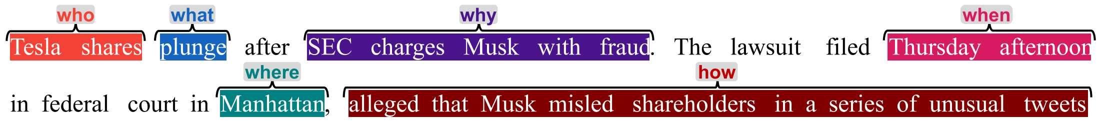


Fig. 3. Example of 5W1H extraction from a document.

Nallapati et al. focused on detecting events along with their dependencies. The authors defined events as clusters of news articles and identified threads of dependent events. Shahaf et al. (2013) presented a concept of information cartography to identify and visualise threads of event-based clusters of news articles and their relationships for a specific user query. A more recent work on event threading by Liu et al. (2020), leveraged event extraction and network analysis to create threads of events in a tree structure. Liu et al. (2020) first proposed an event extraction approach (EventX) that creates a keyword co-occurrence graph to cluster documents with related keywords to determine event clusters. The authors then proposed an event threading approach (StoryForest) that leveraged community detection to link related event clusters in a tree structured event thread. In contrast to event threading (i.e. identifying threads of *dependent* events), in this paper we identify threads about a *particular* event that spreads across multiple documents. However, our proposed approach in this paper is inspired by the components deployed by Nallapati et al. (2004) such as hierarchical clustering using time decay.

To demonstrate the benefits of our information threading approach, we evaluate it in comparison to the three aforementioned related families of methods: (1) document clustering (K-Means), which is generally used in TDT approaches to detect broader events/topics, (2) document threading (k-SDPP) and (3) event extraction (EventX).

3. Proposed approach

In this section, we present our proposed approach, 5W1H-HAC, for identifying coherent threads of information in a large collection of documents. As discussed in Sections 1 and 2, differently from the related tasks in the literature, the primary focus in the Information Threading task is to identify chronological and coherent sequences of documents that are about an event, activity or discussion. Our 5W1H-HAC approach deploys a novel architecture comprising three core components (namely 5W1H Extraction, HAC and Candidate Selection) that collectively enable effective thread identification in an unsupervised setting. Fig. 2 presents the components of our 5W1H-HAC approach. As shown in Fig. 2, the inputs to our approach are all of the documents in a collection as well as the documents' timestamps, which we process through the following components: (1) 5W1H Extraction, which for each document, extracts the text segments that answer each of the 5W1H questions, and concatenates the 5W1H answer segments to form a pseudo-passage that describes the main event that is discussed in the extracted segments of text. (2) HAC, which identifies candidate threads of documents that are related based on the 5W1H pseudo-passages and the amount of time between the creation times of the documents. (3) Candidate Selection, where we finally select the candidates as output threads based on thread coherence and the diversity of information in the thread. In the remainder of this section, we describe in detail these components, namely the 5W1H extraction, the time decay similarity, HAC and the candidate selection components.

3.1. 5W1H extraction

As discussed in Section 1, for each of the documents in the collection, we determine answers to the 5W1H questions (*who*, *what*, *when*, *where*, *why*, and *how*) that can describe the circumstances of an event/activity/discussion that the document is about. Fig. 3, shows an example of answers to the 5W1H questions that can represent the subject (who), temporal characteristics (when), environment (where), cause (why), effect (what) and the method (how). We leverage an existing approach from the literature (Giveme5W1H by Hamborg et al., 2019) for the automatic extraction of the 5W1H questions' answers. Giveme5W1H first identifies candidate text snippets in the documents that represent the action, environment, cause and method of an event. The candidate snippets are then scored by Giveme5W1H to identify the best snippets that can represent the 5W1H answers.

After extracting the 5W1H questions' answers, we concatenate the answers to form a pseudo-passage that describes the main event/activity/discussion in the document (i.e., one pseudo-passage per document). Pseudo-passages are represented as embeddings in a vector space and are used as input to HAC. In our experiments, we compare two types of representations for the pseudo-passages: (1) classic lexical bag-of-words representations, and (2) transformer-based (Vaswani et al., 2017) contextual embeddings.

3.2. Capturing time decay-based similarity

We deploy a tailored similarity function (inspired by Nallapati et al. (2004)) to perform HAC that accounts for: (1) content similarity between the 5W1H pseudo-passages and, (2) the time difference between the creation times of each of the original documents. The content similarity component determines whether a document pair is related based on the *cosine* similarity between the 5W1H pseudo-passage vectors, while the time decay component determines the temporal similarity of the documents (such that a document pair with a larger creation time difference is less similar than a document pair with a smaller time difference). The combined cosine similarity and time decay-based similarity function is defined as follows:

$$sim(d_x, d_y) = cos(p_x, p_y).e^{-\alpha \frac{|t_x - t_y|}{T}} \tag{1}$$

where, p_x and p_y are the pseudo-passage vectors of documents d_x and d_y respectively, cos is the cosine similarity, $e^{-\alpha \frac{|t_x - t_y|}{T}}$ is the normalised time decay component as defined by Nallapati et al. (2004), where t_x and t_y are timestamps of d_x and d_y, respectively, T is the largest time difference between the document timestamps for all documents in the collection, and α is a hyperparameter to factor time decay.

3.3. Hierarchical agglomerative clustering

After extracting the 5W1H questions' answers and representing them as vectorised pseudo-passages, we identify the candidate threads using Hierarchical Agglomerative Clustering (HAC) (Murtagh, 1983). HAC is a widely used text clustering approach that aims to identify hierarchical clusters in a collection by evaluating the hierarchical links between documents in a dendrogram structure. Differently from other popular text clustering methods such as K-Means that simply allocates documents into a fixed number of disjoint clusters, HAC begins with allocating each document as a single cluster and then sequentially combines similar clusters in a bottom-up approach as it moves up in the hierarchy. We use the pseudo-passage vectors and timestamps of the documents as input to HAC. The output from HAC is the clusters (i.e., nodes at a particular hierarchy in the dendrogram) of documents that are related based on the documents' timestamps and the 5W1H pseudo-passages as defined in Eq. (1). We deploy HAC in a complete linkage setting to leverage the similarity function defined in Eq. (1), i.e., the distance D between two clusters, $X\&Y$, is the maximum pairwise distance between all document pairs of $X\&Y$, defined as:

$$D_{complete}(X, Y) = \max_{x \in X, y \in Y} (1 - sim(x, y)) \tag{2}$$

We evaluate the effectiveness of our proposed setting (i.e., using a time decay-based complete linkage) when deploying HAC for information threading compared to the popular Ward linkage algorithm (Ward, 1963). Ward focuses on minimising the variance of the clusters being combined by computing the error sum of squares (ESS) of the clusters. The linkage distance D in the ward algorithm between two clusters, $X\&Y$, is defined as the increase in ESS of the combined cluster XY compared to the ESS of the individual clusters:

$$D_{ward}(X, Y) = ESS(XY) - (ESS(X) + ESS(Y)) \tag{3}$$

We treat each output HAC cluster as a set of documents that are associated with a potential thread. In particular, for each of the output HAC clusters, we leverage the creation timestamp of documents in a cluster to form a chronological sequence of the documents as a candidate thread. We then select coherent information threads from the pool of candidate threads as described in the following section.

We argue that hierarchical clustering is well suited for the information threading task, since the dendrogram hierarchies can naturally represent the following association: documents→ threads (about events) → higher-level topics or subject-domains. In addition, the number of clusters (i.e., candidate threads) in information threading is considerably higher than in a general clustering task such as identifying topical clusters (e.g. 8 news topics vs 27,681 thread labels for the NewSHead (Gu et al., 2020) articles, which we describe in Section 4). Therefore, the bottom-up algorithm of HAC for moving up in the dendrogram hierarchies can be stopped much earlier in the case of threading after exploring a desired number of clusters. We analyse the efficiency of HAC compared to the popular K-Means clustering in Section 7.3, where we show that HAC is markedly more efficient than K-Means for the information threading task.

3.4. Selecting information threads from candidates

After generating the candidate information threads using HAC, we select the candidate threads that are estimated to be the most coherent and to provide diverse information about the event/activity/discussion that they discuss. We now discuss the process of selecting such coherent and diverse information threads in an unsupervised setting. To determine coherence, we use a topic coherence metric (i.e., the C_V metric by Röder & Hinneburg, 2015) that is used in the topic-modelling approaches (as surveyed by Churchill & Singh, 2022; Zhao et al., 2021) to measure the extent to which the generated topics are human interpretable. In particular, we leverage this definition of coherence from the topic-modelling literature for information threading to determine the human interpretability of the generated threads based on whether the documents in the thread discuss the same event. In addition to coherence, it is also important to measure information diversity so that the candidates that are selected to be threads do not contain a lot of repeated information. For example, news collections that contain articles from multiple news agencies (e.g. NewSHead) can have duplicate information in multiple articles about the same event, which can be grouped as less informative threads of repeated information. To measure the information diversity of a thread, we can compute the mean of the KL Divergence (Kullback & Leibler, 1951) scores when, for each document in a thread, a document is held-out and the KL Divergence is computed between the probability distributions of the words in the held-out document and the words in the rest of the documents in the thread. However, the C_V coherence metric and the held-out KL divergence measure can be expensive to compute on a large collection with a large number of potential threads. Therefore, using a subset of the larger collection, we determine the minimum and maximum threshold parameter values for three different measures that we then use as an estimate of the coherence and diversity of a thread. In particular, we first sample a subset of the documents from the larger collection to form multiple (small) sets of documents, which we refer to as the validation sets (we provide details about how we sample these sets in Section 4.1). We then deploy HAC on each of the validation sets individually and calculate the mean coherence (using C_V) and the mean diversity (using the held-out KL divergence) from all of the candidate threads in the validation sets. The mean coherence and mean diversity scores can then be used to optimise the parameter values of the three measures such that the candidates that have the maximum coherence and diversity are selected as threads. The three measures that we use to estimate the coherence and diversity of threads, and to finally select threads from the larger collection, are as follows:

1. The minimum and maximum acceptable number of documents in a thread $\mathbb{T}$ (i.e., the thread length $|\mathbb{T}|$),
2. The minimum and maximum acceptable time period, $\mathbb{T}_{span}$, between the creation dates of the first and last documents in a candidate thread
3. The minimum and maximum acceptable 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, $d_x \in \mathbb{T}$, defined as:

$$\mathbb{T}_{MPDCS} = \frac{1}{|\mathbb{T}|-1} \cdot \sum_{x=1}^{|\mathbb{T}|-1} cos(d_x, d_{x+1}) \tag{4}$$

We use a multi-objective optimisation to identify the best combination of (minimum and maximum) parameter values for each of the measures to select the threads that maximise the mean coherence (ζ; computed using C_v), mean diversity (δ; computed using the held-out KL divergence) and the number of selected threads (n) on S number of validation sets. In particular, for each parameter combination, θ, in the set of all parameter combinations, Θ, we compute the mean of ζ_θ, δ_θ and n_θ as $x_\theta = \frac{1}{s} \cdot \sum_{i=1}^{S} x_\theta^i$ (where $x \in \{\zeta, \delta, n\}$). We then identify the set of non-dominated solutions from Θ (aka Pareto optimal solutions, $\Theta_{NDS} \subset \Theta$) using the NSGA-II algorithm (Deb, Pratap, Agarwal, & Meyarivan, 2002). We further select the best combination of parameters, $\theta' \in \Theta_{NDS}$, for which we observe a maximum of the individual standardised[1] scores $\hat{\zeta}$, $\hat{\delta}$ and $\hat{n}$ with a minimum difference between the individual scores, defined as:

$$\theta' = \underset{\theta \in \Theta_{NDS}}{\operatorname{argmax}} \frac{\hat{\zeta}_\theta + \hat{\delta}_\theta + \hat{n}_\theta}{|\hat{\zeta}_\theta - \hat{\delta}_\theta| + |\hat{\zeta}_\theta - \hat{n}_\theta| + |\hat{\delta}_\theta - \hat{n}_\theta|} \tag{5}$$

Overall, θ' is the best estimated combination of the threshold parameters for $|\mathbb{T}|, \mathbb{T}_{span}$ and $\mathbb{T}_{MPDCS}$, which we use when selecting the final output threads from the candidates (we provide details of the set Θ that we use to identify θ' for our experiments in Section 4.3)

When selecting the final threads that are to be output by our 5W1H-HAC information threading approach, we select the threads that are within the minimum and maximum acceptable threshold limits of each of the aforementioned measures. For example, if the threshold limit of the threads' lengths is [3,10], then a candidate thread $\mathbb{T}$ is selected only if, $3 \leq |\mathbb{T}| \leq 10$.

4. Experimental setup

We now describe our experimental setup for the offline evaluation (Section 5) as well as for the conducted user study (Section 6). In particular, we describe: (1) the document collection for evaluating the threading approaches, (2) the baseline approaches that we evaluate, and (3) the configurations of our proposed 5W1H-HAC approach.

4.1. Dataset

There are very limited test collections available for the evaluation of information threads that describe an event, activity or a discussion. Moreover, previous related work on document and event threading (e.g. Gillenwater et al., 2012; Nallapati et al., 2004) often evaluate their approaches using manual annotations, which are not publicly available. As mentioned in Section 1, the news domain can be one of the direct applications of information threads. Therefore, for our offline evaluation and our user study we experiment with test collections that comprise news articles and labels about the main event/activity/discussion described in the articles. In particular, we use the NewSHead (Gu et al., 2020) and the Multi-News (Fabbri et al., 2019) test collections.

[1] We standardise the scores by removing the mean and scaling to unit variance.

- **NewSHead** (Gu et al., 2020): The NewSHead collection contains URLs to 932,571 news articles that were published by various news agencies between May 2018 and May 2019. We could only crawl a subset of 112,794 news articles from the URLs specified in the NewSHead collection. We focus our experiments on this subset of available news articles. The NewSHead collection also contains news story labels, where a story label corresponds to a group of news articles about the same event. The 112,794 NewSHead articles that are used in our experiments are associated with 95,786 story labels.
  The NewSHead articles are often associated with more than one story labels, i.e., the stories can be overlapping sets of articles. For our evaluation of information threading approaches, we perform a union of such overlapping article sets that are each corresponding to a story label, and refer to the union sets as the ground truth thread labels. For example, the NewSHead article shown in Fig. 3 is related to multiple story labels such as "Elon Musk charged with SEC", "Tesla charged with fraud", "SEC charges Tesla". We combine all the articles that are associated with the three aforementioned example stories into a single thread ground truth label. Overall, we created 27,681 ground truth thread labels for the NewSHead articles.
- **Multi-News** (Fabbri et al., 2019): The Multi-News collection contains summaries of 56,216 news events and their corresponding news articles, i.e., multiple news articles associated to an event. We leverage this association of news articles and the events that the articles describe as ground-truth for the information threading task. However, since Multi-News does not contain the article creation timestamps (as required by our evaluated methods), we crawl the original articles using the URLs provided by the authors of the collection. Next, we select the events that are associated to at least three crawled articles with valid timestamps for evaluating our information threading approach. Overall, we were able to find 32,249 news articles with valid timestamps, which are associated with 9,378 events (i.e., the ground truth thread labels).

Similarly to Gillenwater et al. (2012), to reduce the time taken to run our experiments, we split the NewSHead collection uniformly into three groups (37,598 articles each) based on the article creation time, which we refer to as the NewsHead test sets. We deploy each of the evaluated approaches on the three test sets separately, and report evaluation results for the identified threads across all the three test sets. In the case of the Multi-News collection, due to the relatively small number of articles (i.e, 32,249) compared to NewSHead, we consider the entire Multi-News collection as a single test set.

We further create three smaller subsets (i.e., the validation sets; $S = 3$) from the test sets for parameter tuning, where articles in each validation set are associated with 1000 randomly sampled thread labels. Since, we do not use any ground truth thread labels for parameter tuning, the overlap between the test and validation sets cannot lead to any overfitting.

4.2. Baselines

We evaluate the effectiveness of our proposed 5W1H-HAC information threading approach (c.f. Section 3) compared to the following three baselines from the literature:

- **K-Means** (Lloyd, 1982; MacQueen, 1967): The first approach that we compare against is the K-Means document clustering approach. We perform K-Means clustering on the articles in the three test sets using their TF-IDF vectors projected onto a 200-D dense space by latent semantic analysis (LSA). We use the default scikit-learn (Pedregosa et al., 2011) implementation for deploying K-Means, TF-IDF vectorisation and LSA. Since, K-Means require a fixed number of clusters, we set the number of clusters as the total number of thread labels in each of the test sets. Finally, we select the output K-Means candidate clusters based on the same criteria as described in Section 3.4 (i.e., $|\mathbb{T}|, \mathbb{T}_{span}$ and $\mathbb{T}_{MPDCS}$).
- **k-SDPP** (Gillenwater et al., 2012): The second approach that we compare against is the k-SDPP document threading approach. We use a publicly available implementation of SDPP sampling (Kulesza & Taskar, 2010), and deploy TF-IDF term features to create the document graph as described in Gillenwater et al. (2012). Since k-SDPP returns threads of a fixed length (i.e., $|\mathbb{T}|$), we specify $|\mathbb{T}| = 4$ for NewSHead and $|\mathbb{T}| = 3$ for Multi-News based on the mean thread length in each collection, respectively. Moreover, since k-SDPP samples a small number of threads in a single execution, we deploy k-SDPP for 200 executions with a sample size of 50 threads on each of the three test sets, e.g., $200 * 50 * 3 =$ 30,000 candidate threads (i.e., $30,000 \approx 27,681$ ground truth threads in NewSHead). We discard any duplicate candidate threads, before evaluating them collectively.
- **EventX** (Liu et al., 2020): The third approach that we compare against is the EventX event extraction approach, using the publicly available implementation (Liu et al., 2020). The EventX approach requires the articles and their topics as input. Therefore, based on the 8 NewSHead topics presented by Gu et al. (2020), we acquire topic labels for the NewSHead and Multi-News articles using a news topic classifier. In particular, we fine-tune the distilBERT (Sanh, Debut, Chaumond, & Wolf, 2019) model to classify news topics (e.g. Politics or Sports) using the publicly available News Category dataset.[2] We use this distilBERT classifier to infer the topics of the NewSHead and Multi-News articles.

4.3. 5W1H-HAC

We now discuss the implementation details of our proposed 5W1H-HAC threading approach along with the different configurations that we evaluate for this proposed approach.

[2] News Category Dataset: https://www.kaggle.com/datasets/rmisra/news-category-dataset.

Table 1
Sets used for tuning 5W1H-HAC parameter values.

Parameter	Set	
α	$\{10^i \; \forall \; -4 \le i \le 4;\ \text{step} = 1\}$	
$x \le \lvert\mathbb{T}\rvert \le y$	$\{x, y\} \in \{\{3, i\} \; \forall \; 10 \le i \le 100;\ \text{step} = 10\}$	
$x \le \mathbb{T}_{span} \le y$	$\{x, y\} \in \{\{0, i\} \; \forall \; 30 \le i \le 360;\ \text{step} = 30\}$	(NewSHead)
	$\{x, y\} \in \{\{0, 360 * i\} \; \forall \; i \in \{1/12, 1/4, 1/2, 1, 2, 3, 4, 5\}\}$	(Multi-News)
$x \le \mathbb{T}_{MPDCS} \le y$	$\{x, y\} \in \{\{0 + i, 1 - i\} \; \forall \; 0 \le i \le 0.4;\ \text{step} = 0.1\}$	

- **5W1H-Extraction**: We use the publicly available implementation of Giveme5W1H (Hamborg et al., 2019) for 5W1H extraction from the news articles, and concatenate the extracted answers to the 5W1H questions to form a pseudo-passage for each article. As mentioned in Section 3.1, we compare lexical bag-of-words and contextual embedding representations of the pseudo-passages. In particular, we evaluate three different representations of the pseudo-passages for generating the threads: (1) TF-IDF term features (Pedregosa et al., 2011), and two variants of contextual embeddings, namely: (2) *all-miniLM-L6-v2* and (3) *all-distilRoBERTa-v1* from the Sentence Transformer Library (Reimers & Gurevych, 2019).
- **Hierarchical Agglomerative Clustering** (HAC): We deploy HAC using the scikit-learn (Pedregosa et al., 2011) implementation. Similar to the K-Means baseline, we use the total number of thread labels in each of the test sets as the number of clusters for HAC.
- **Configurations**: Based on the different combinations of the pseudo-passage representations, i.e., TF-IDF, miniLM or distilRoBERTa, and the deployed HAC linkage strategy, i.e., time decay-based complete linkage (TD) or Ward linkage (W) (c.f. Section 3.3), we denote the different configurations of our proposed approach as 5W1H-HAC-<*Linkage*>-<*Features*> (e.g., 5W1H-HAC-TD-miniLM refers to the time decay-based complete linkage and miniLM representations).
- **Parameters**: Table 1 presents the sets that we use to tune the parameters specified in Section 3, i.e., the time decay factor (α) and the threshold limit parameters for the estimated coherence and diversity measures ($\lvert\mathbb{T}\rvert$, $\mathbb{T}_{span}$ and $\mathbb{T}_{MPDCS}$). We tune the parameters for the various 5W1H-HAC configurations based on their average effectiveness on the three validation sets of both the NewSHead and Multi-News collections. Recall that we tune the parameters without using the ground-truth thread labels, i.e., we only use the documents in the collection to estimate the best parameters for the proposed approach.

5. Offline evaluation

In this section, we present the offline evaluation of our proposed 5W1H-HAC approach in comparison to the document clustering (K-Means), document threading (k-SDPP) and event extraction (EventX) baselines. We first discuss our evaluation metrics in Section 5.1 before presenting the results in Section 5.2. Our offline evaluation aims to answer the following three research questions:

RQ1: Is our proposed 5W1H-HAC information threading approach more effective for generating good quality information threads than the existing approaches from the literature?
RQ2: Are contextual embeddings more effective than TF-IDF vectors for representing the 5W1H pseudo-passages?
RQ3: Does deploying our proposed time decay similarity function in our 5W1H-HAC threading approach increase the quality of the generated threads, compared to the Ward linkage strategy?

5.1. Evaluation metrics

Since threads are in general small document clusters, we measure the quality of the threads that are generated by the different evaluated approaches using the following two cluster quality metrics:

- **Homogeneity Score** (h) (Rosenberg & Hirschberg, 2007), which measures the extent to which the resulting clusters meet the homogeneity criteria, i.e, whether data points in the clusters are members of a single true class.
- **Normalised Mutual Information** (NMI) (Cai, He, & Han, 2005), which measures the uncertainty in the model in assigning a document to a cluster.

Compared to other general cluster quality metrics such as clustering accuracy (Xie, Girshick, & Farhadi, 2016) and pairwise F_1 (Nallapati et al., 2004), both h and NMI are computationally inexpensive. Therefore, these metrics are well suited for evaluating thread quality for large collections such as NewSHead.

As mentioned in Section 4.1, all of the articles in the NewSHead and Multi-News collections have an associated thread ground truth label. However, our proposed approach and the baseline approaches do not necessarily select all of the articles to be part of a generated thread. This results in two possible scenarios for evaluating the effectiveness of the threading approaches. Firstly, we evaluate the effectiveness of the approaches in terms of h and NMI using only the ground truth labels of the NewSHead and Multi-News articles that *are selected* to be part of an information thread (we refer to this scenario as Generated Threads). Secondly, since the number of documents identified as part of the threads is an important factor, we evaluate the h and NMI of the approaches using the ground truth labels of *all* of the articles in the NewSHead and Multi-News collections, respectively. This provides a measure of overall effectiveness of a threading approach (we refer to this scenario as Overall Performance).

Table 2
Comparative thread quality results (higher scores are better). TD refers to time decay similarity and W to Ward linkage.

	Configurations	NewSHead				Multi-news			
		Generated threads		Overall performance		Generated threads		Overall performance	
		h	NMI	h	NMI	h	NMI	h	NMI
Baseline	K-Means	0.6458	0.7848	0.0001	0.0003	0.7447	0.8537	0.0010	0.0021
	k-SDPP	0.8819	0.8962	0.1079	0.1908	**0.8911**	0.8979	0.1318	0.2273
	EventX	0.8241	0.8832	0.1415	0.2405	0.8139	0.8808	0.1326	0.2274
5W1H-HAC	W-TFIDF	0.8366	0.8676	0.5043	0.6286	0.8337	0.8726	0.3903	0.5294
	W-miniLM	0.8947	0.9129	0.5937	0.7157	0.8743	0.8989	0.5989	0.7121
	W-distilRoBERTa	0.8918	0.9098	0.5812	0.7053	0.8718	0.8963	0.6021	0.7139
	TD-TFIDF	0.8508	0.8856	0.5063	0.6369	0.8211	0.8582	0.6215	0.7195
	TD-miniLM	**0.9144**	**0.9348**	**0.6329**	**0.7537**	0.8827	**0.9093**	**0.7165**	**0.8008**
	TD-distilRoBERTa	0.9106	0.9318	0.6082	0.7350	0.8803	0.9080	0.7112	0.7978

5.2. Results

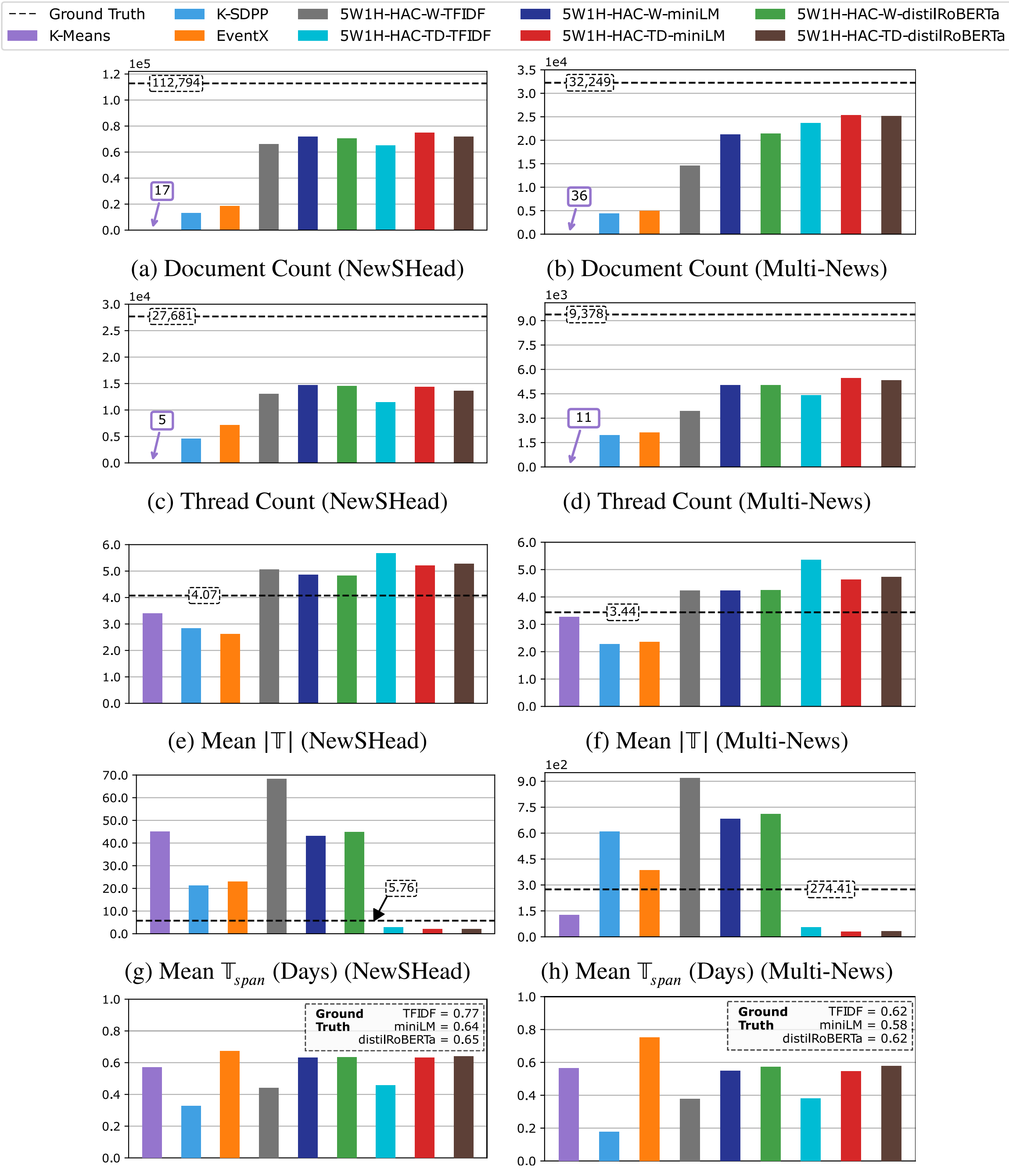
We now discuss the results of our offline evaluation. Table 2 presents the results of our experiments to evaluate the quality of the threads under the two setups discussed in Section 5.1 (i.e., Generated Threads and Overall Performance), in terms of Homogeneity (h) and Normalised Mutual Information (NMI). As shown in Fig. 4, we also report the number of documents that are identified as being part of a thread, the number of generated threads, the mean thread length ($|\mathbb{T}|$), the mean thread time span ($\mathbb{T}_{span}$) and the mean pairwise document cosine similarity ($\mathbb{T}_{MPDCS}$) of the threads.

Firstly, addressing **RQ1**, we observe from Table 2 that all of the configurations of our proposed 5W1H-HAC approach markedly outperform K-Means, k-SDPP and EventX under the Overall Performance setup on both the NewsHead and Multi-News collections (e.g., NewSHead; TD-miniLM: 0.7537 NMI vs K-Means: 0.0003 NMI, k-SDPP: 0.1908 NMI & EventX: 0.2405 NMI). Under the Generated Threads setup, we first observe that the threads generated by K-Means achieve the lowest h and NMI. Moreover, on the NewSHead collection, all of the 5W1H-HAC configurations, excepting the TFIDF configurations, generate threads that are of higher quality than those from the k-SDPP and EventX approaches in terms of h and NMI scores. In addition, on the Multi-News collection, all the time decay (TD) 5W1H-HAC configurations (excepting TFIDF) outperform k-SDPP and EventX in terms of NMI. However, k-SDPP achieves slightly higher homogeneity (h) than the 5W1H-HAC configurations on Multi-News (e.g. TD-miniLM: 0.8827 NMI vs k-SDPP: 0.8911). Even though the improvements by the 5W1H-HAC approaches under the Generated Threads setup are comparatively smaller than the improvements under the Overall Performance setup, the number of generated threads (and the number of documents associated with the threads) is noticeably higher compared to the baseline approaches as shown in Figs. 4(a), 4(b), 4(c) and 4(d). This improvement in the number of generated threads along with the improvement in the quality of the threads (under both setups), shows that our proposed 5W1H-HAC approach can identify quality threads that comprise a majority of the documents in the NewSHead collection (collectively on the three test sets specified in Section 4.1) and the Multi-News collection. Therefore, in response to RQ1, we conclude that our 5W1H-HAC information threading approach is indeed effective since it markedly improves the number of, and the quality of, the generated threads compared to the document clustering (K-Means), document threading (k-SDPP) and event extraction (EventX) approaches.

Moving on to **RQ2**, from Table 2 we observe that all of the configurations of our 5W1H-HAC approach that deploy contextual embedding representations of the 5W1H pseudo-passages (i.e., miniLM & distilRoBERTa) outperform the configurations that deploys the TF-IDF representations, in terms of both h and NMI. These improvements with contextual embeddings are consistent when either the Ward linkage “W” or the time decay-based similarity “TD” are deployed. Therefore, in response to RQ2, we conclude that leveraging the contextual similarity of the 5W1H pseudo-passages is notably more effective than deploying the classic TF-IDF representations.

Lastly addressing **RQ3**, from Table 2 we observe that the time decay-based HAC configuration, TD-miniLM, is the most effective, under the Overall Performance Setup (e.g. NewSHead: 0.6329 h and 0.7537 NMI). Moreover, we can see from Figs. 4(a) and 4(b), that 5W1H-HAC-TD-miniLM identifies the highest number of documents that are associated with the threads (i.e., 66.28% and 78.68% of the total NewSHead and Multi-News documents, respectively) with the best h and NMI scores, under the Generated Threads setup (e.g. NewSHead: 0.9144 & 0.9348). Both the miniLM and distilRoBERTa variants of the “TD” configuration outperform the respective variants in the ward “W” configuration (e.g., miniLM +6.6% h & +5.3% NMI on the NewSHead collection). In addition, as shown in Figs. 4(g) and 4(h), the mean time span of threads generated by the “TD” configurations is the closest to true time span (e.g., NewSHead Ground Truth: 5.76 days vs TD-TFIDF: 2.82 vs TD-miniLM: 2.07 days & TD-distilRoBERTa: 2.11 days) compared to the ward “W” 5W1H-HAC configurations. Therefore in response in RQ3, we conclude that 5W1H-HAC with a time decay-based similarity function is more effective than the Ward linkage strategy, and is overall the most effective information threading approach among those that we have evaluated.

We select the best 5W1H-HAC configuration (i.e., TD-miniLM) for our following user study. Due to the markedly low number of generated threads by K-Means (e.g., only 5 NewSHead threads, see Fig. 4(c)), we only select the k-SDPP and EventX baselines for our following user study.



(a) Document Count (NewSHead) (b) Document Count (Multi-News)

(c) Thread Count (NewSHead) (d) Thread Count (Multi-News)

(e) Mean $|\mathbb{T}|$ (NewSHead) (f) Mean $|\mathbb{T}|$ (Multi-News)

(g) Mean $\mathbb{T}_{span}$ (Days) (NewSHead) (h) Mean $\mathbb{T}_{span}$ (Days) (Multi-News)

(i) Mean $\mathbb{T}_{MPDCS}$ (NewSHead) (j) Mean $\mathbb{T}_{MPDCS}$ (Multi-News)

Fig. 4. Comparison of various statistics of information threads that are generated from the evaluated approaches.

6. User study

The offline evaluation in Section 5 was limited to evaluating the effectiveness of a threading approach only in terms of the documents that the threads contain, compared to the ground truth of the test collection. However, an information thread is not just a cluster of documents but primarily a coherent chronological *sequence* of related documents. Therefore, it is essential to evaluate whether the threads provide meaningful sequences of information about an event/activity/discussion to the human users. In this section, we conduct a user study to evaluate the effectiveness of our proposed 5W1H-HAC information threading approach, compared

Table 3
Participant groups for the user study based on a balanced Latin square counterbalancing of the pairs of approaches (*right*) and the test conditions (*left*).

Id	Method#1	Method#2
A	k-SDPP	5w1h-HAC
B	5w1h-HAC	k-SDPP
C	EventX	5w1h-HAC
D	5w1h-HAC	EventX
E	k-SDPP	EventX
F	EventX	k-SDPP

Order →	1st	2nd	3rd	4th	5th	6th
Group 1	A	B	F	C	E	D
Group 2	B	C	A	D	F	E
Group 3	C	D	B	E	A	F
Group 4	D	E	C	F	B	A
Group 5	E	F	D	A	C	B
Group 6	F	A	E	B	D	C

to the k-SDPP and EventX approaches from the literature. We selected the 5W1H-HAC-TD-miniLM configuration to evaluate in our user study since it was found to be the best evaluated configuration in Section 5.2. For the user study, we conduct a pairwise evaluation of the threads that are generated from the aforementioned three approaches to identify the participants' preferences in terms of the coherence, diversity of information, and chronological correctness of the threads, as well as the participants' overall preferences. Our user study aims to answer the following research question:

RQ4: Do the human users prefer the threads identified by our 5W1H-HAC approach compared to the methods from the literature?

6.1. Experiment design

Our user study follows a within-subject design, i.e., all of the participants were presented with all of the three possible pairs of threading approaches: 5W1H-HAC vs EventX, 5W1H-HAC vs k-SDPP, and k-SDPP vs EventX. In particular, the participants were presented with 6 pairs of threads (two from each of the three pairs of approaches), where both of the threads in a pair describe the same event. Table 3 shows the 6 approach-thread pairs. We used balanced Latin square counterbalancing to create a participant group respective to each of the approach-thread pairs. For each of the pairs of threads, the participants were asked to select the thread that they preferred overall based on the sequence of information. Additionally, the participants were asked to rate each of the threads individually, with respect to: (1) how many articles in the thread belong to the same event, i.e. coherence, (2) how many articles in a thread provide diverse information about the same event, i.e. diversity, and (3) how many articles in a thread follow the correct chronological order as per the true chronology of the information presented in the thread. We captured the participants' ratings of the aforementioned three inputs for each thread on a 4-point likert scale with the following options: (1) None of the articles, (2) Some of the articles, (3) Most of the articles and (4) All of the articles. The 4-point likert scale is well-suited as per the number of articles we restrict in each sample thread (i.e. 4), as discussed in the next section. To reduce the time and complexity of reading large articles, we present the participants with only the titles of the articles from the threads.

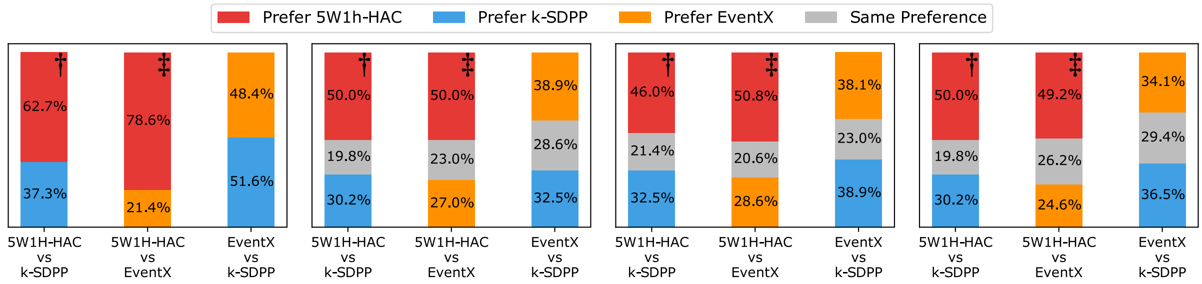
6.2. Selecting pairs of threads

To select the pairs of threads to present to the participants of our user study, we performed a controlled sampling of 6 pairs of threads that were generated by the evaluated approaches from the NewSHead collection, i.e. two pairs of threads for each of the three pairwise combinations between 5W1H-HAC, k-SDPP and EventX (shown in Table 3). We controlled the number of documents in each of the sampled threads to be exactly 4 (i.e., $|\mathbb{T}| = 4$) based on the mean thread length in the NewSHead collection. To help the participants in their comparisons of two different threads in a pair, we selected the pairs where the majority of the documents in both the threads discuss the same event. To select such pairs of threads, we leverage the ground truth thread labels that we presented in Section 4.1. In particular, we used the ground truth thread label that is associated with the majority of documents in a generated thread as the gold label of the generated thread. From all possible pairs of the generated threads, we then selected the pairs where each thread in a pair is associated with the same gold label. To ensure a fair comparison of the threads generated by two different methods in a pair, we ranked the selected pairs of threads based on two scores: (1) the mean pairwise document cosine similarity of a thread, i.e., $\mathbb{T}_{MPDCS}$ (Eq. (4)), and (2) the precision score of a thread $\mathbb{T}_{prec}$, which is the ratio of the number of documents associated with the gold label t' to the total number of documents in the thread $\mathbb{T}$, i.e., $\mathbb{T}_{prec} = |\mathbb{T}_{t'}|/|\mathbb{T}|$. For the aforementioned scores ($\psi \in \{prec, MPDCS\}$), we deploy a gain function $\mathcal{G}_{\psi}$ that favours the pairs of threads with the higher individual scores and the lower trade-off between the scores of threads, $\mathbb{A}\&\mathbb{B}$, in a pair, defined as follows:

$$\mathcal{G}_{\psi}^{\mathbb{A}\mathbb{B}} = \mathbb{A}_{\psi}.\mathbb{B}_{\psi} - abs(\mathbb{A}_{\psi} - \mathbb{B}_{\psi}) \tag{6}$$

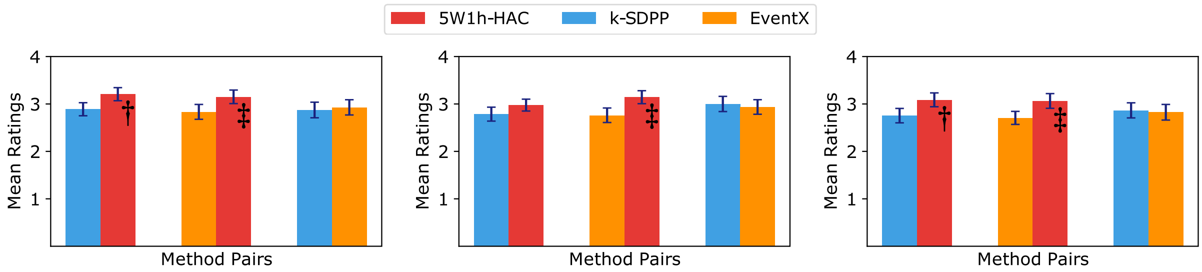
In a set $\mathbb{C}$ of all the selected pairs of threads, we sort the pairs of threads first based on $\mathcal{G}_{prec}$ and then based on $\mathcal{G}_{MPDCS}$, to find the top-n pairs defined as follows:

$$sample(\mathbb{C}) = \operatorname*{argsort}_{c \in \mathbb{C}} \left(-\mathcal{G}_{prec}^{c}, -\mathcal{G}_{MPDCS}^{c}; n = 2 \right) \tag{7}$$



(a) Overall Preference (b) Coherence (c) Diversity of Information (d) Chronological Correctness

Fig. 5. Pairwise participants' preferences of the threading methods. Statistically significant (chi-square test, $p < 0.05$) proportions of preferences for the 5W1H-HAC threads are denoted by "†" & "‡" wrt k-SDPP & EventX, respectively.



(a) Coherence (b) Diversity of Information (c) Chronological Correctness

Fig. 6. Mean participants' ratings of the threading methods. Statistically significant (t-Test, $p < 0.05$) improvements in ratings for the 5W1H-HAC threads are denoted by "†" & "‡" compared to k-SDPP & EventX respectively.

6.3. Participant recruitment

We recruited 63 participants using the MTurk[3] crowdsourcing platform. We restricted the participants to be aged 18+ years and from countries where English is their first language. The 63 participants were assigned uniformly across the 6 participant groups as shown in Table 3. The participants' recruitment criteria have been approved by our organisation's ethical board.

6.4. Evaluation criteria

We evaluate the effectiveness of the three threading approaches (5W1H-HAC, k-SDPP and EventX) based on the participants' preferences and ratings of the threads generated from each approach. *First*, we evaluate the proportion of participants that prefer a threading method in a pair. Since we capture coherence, diversity and chronological correctness of the threads as ratings, we consider the highest rated thread in a pair as the preferred thread. We use the chi-square goodness-of-fit test to measure statistical significance for the observed proportion of participants preferring a threading method, and we select $p < 0.05$ as our significance threshold. We report the observed power, chi-square (χ^2) statistics and Cohen's w effect size for the chi-square goodness-of-fit tests. *Second*, we evaluate the participants' ratings of a threading method in a pair, i.e., how good the participants rated a thread from a threading method. For each of the three rating criteria (i.e., coherence, diversity and chronological correctness), we compute the mean of the participants' ratings of a threading method in a pair. We use paired-samples t-Test to measure statistical significance between the mean participants' ratings of each of the threading methods in a pair. We select $p < 0.05$ as our significance threshold, and report the observed power and Cohen's d effect size for the t-Test.

6.5. User study results

We now discuss the results of our user study in Section 6. Fig. 5 shows the percentages of the participants' preferences in the pairwise comparison of the three threading approaches (5W1H-HAC, k-SDPP and EventX). Fig. 6 shows the mean participants' ratings for the threads generated by the three evaluated approaches. In Figs. 5 & 6, statistically significant improvements ($p < 0.05$) compared to the k-SDPP and EventX are denoted as "†" & "‡", respectively. Table 4 presents the results of the statistical significance

[3] https://www.mturk.com/

Table 4
Participants' preferences (Chi-square test) and the mean participants' ratings (t-Test). χ is the chi-square statistics, df is the degree of freedom, p is the significance and "bold" represents statistical significant result at $p < 0.05$.

Criteria	Configuration			Chi-Square goodness-of-fit test (preference)				Paired samples t-Test (ratings)		
				$\chi^2(df)$	Cohen's w	p	Power	Cohen's d	p	Power
Overall	5W1H-HAC	vs	k-SDPP	**8.127** (1)	**0.254**	**0.004**	**81.36%**	–	–	–
	5W1H-HAC	vs	EventX	**41.143** (1)	**0.571**	**< 0.001**	**100.00%**	–	–	–
	k-SDPP	vs	EventX	0.127 (1)	0.032	0.722	6.49%	–	–	–
Coherence	5W1H-HAC	vs	k-SDPP	**17.762** (2)	**0.375**	**< 0.001**	**97.25%**	**0.268**	**0.003**	**84.80%**
	5W1H-HAC	vs	EventX	**16.048** (2)	**0.357**	**< 0.001**	**95.73%**	**0.262**	**0.004**	**83.10%**
	k-SDPP	vs	EventX	2.048 (2)	0.127	0.359	22.86%	0.044	0.620	7.80%
Diversity	5W1H-HAC	vs	k-SDPP	**11.476** (2)	**0.302**	**< 0.001**	**86.84%**	0.165	0.067	45.00%
	5W1H-HAC	vs	EventX	**18.476** (2)	**0.383**	**< 0.001**	**97.76%**	**0.313**	**0.001**	**93.60%**
	k-SDPP	vs	EventX	6.048 (2)	0.219	0.050	58.73%	0.045	0.615	7.90%
Chronological Correctness	5W1H-HAC	vs	k-SDPP	**17.762** (2)	**0.375**	**< 0.001**	**97.25%**	**0.272**	**0.003**	**85.70%**
	5W1H-HAC	vs	EventX	**14.333** (2)	**0.337**	**0.001**	**93.34%**	**0.309**	**0.001**	**93.10%**
	k-SDPP	vs	EventX	1.000 (2)	0.089	0.607	13.25%	0.031	0.727	6.40%

tests, i.e., the chi-square goodness-of-fit test and the paired samples t-Test when comparing the participants' preferences and ratings respectively.

Firstly evaluating the participants' preferences, Fig. 5 shows that participants significantly (chi-square test, $p < 0.05$) prefer the 5W1H-HAC threads compared to the threads from both k-SDPP and EventX across all four of the criteria: overall preference, coherence, diversity and chronological correctness. We further inspected the participants' preferences of k-SDPP and EventX. We observe that, k-SDPP is preferred over EventX in terms of diversity and chronological correctness, while EventX is preferred in terms of coherence. We discuss this observation in Section 8. However, as shown in Table 4, the comparisons between preferences for k-SDPP and EventX are not significant. Secondly, we evaluate the mean participants' ratings. Fig. 6 shows that the participants provided higher ratings for the 5W1H-HAC threads compared to both the k-SDPP and EventX threads across all of the three criteria, i.e., coherence (+10.99% & +11.20%), diversity (+6.84% & +13.79%) and chronological correctness (+12.10% & +13.20%). According to the paired samples t-Test results in Table 4, the participants rated the 5W1H-HAC threads significantly ($p < 0.05$) higher compared to the EventX threads, in terms of coherence, diversity and chronological correctness. Moreover, compared to the k-SDPP threads, the participants rated the 5W1H-HAC threads as significantly more coherent and chronologically correct. Furthermore, comparing the ratings of k-SDPP and EventX, we observe that k-SDPP is rated higher compared to EventX in terms of diversity and chronological correctness, while EventX is rated higher over k-SDPP in terms of coherence. However from Table 4, we observe that the differences between the participants' ratings for k-SDPP and EventX are not significant.

In response to **RQ4**, we conclude that the threads generated by 5W1H-HAC are indeed significantly (chi-square test, $p < 0.05$) preferred by the participants, compared to the threads from k-SDPP and EventX. The participants also rated the threads from 5W1H-HAC significantly higher (t-test, $p < 0.05$) in terms of coherence, diversity and chronological correctness compared to EventX, and in terms of coherence and chronological correctness compared to k-SDPP.

7. Analysis

In this section, we provide an analysis of our offline evaluation and user study.

7.1. Qualitative analysis

Fig. 7 presents three randomly sampled threads that are generated by our 5W1H-HAC approach (TD-miniLM configuration). Thread#1 presents three news articles describing the origin, process and outcome of the event *Trump's fight over a closed GM plant*. Thread#2 discusses related news articles that mention the activity *stranded aircraft taking off from Iraq*. Thread#3 presents the origin and follow-up stories of a discussion about *Hurricane Maria death toll*. Even though some of the articles provide repeated information (e.g., the last two articles in Thread#2 and the first two articles in Thread#3), overall the threads present coherent and chronological sequences of related information, as per the definition of information threads that we presented in Section 1.

7.2. Role of time decay component

It is also important to analyse the role of the time decay component in improving the quality of the threads, and how to select the right value for the α parameter that factors the time decay component in the HAC similarity function (Eq. (1)). Fig. 8 shows the effect of the time decay factor α on the thread quality metrics (h and NMI) and the similarity scores in HAC. From Fig. 8(a), we observe that the thread quality scores improve when $\alpha > 0.1$ and peak at $\alpha = 10$ before rapidly declining when $\alpha > 100$. We investigate this trend of thread quality by analysing the individual cosine (cos) and time decay (TD) similarity scores along with the combined $cos * \text{TD}$ similarity score from Eq. (1). Recall that, cos is the cosine similarity of the 5W1H pseudo-passages and TD

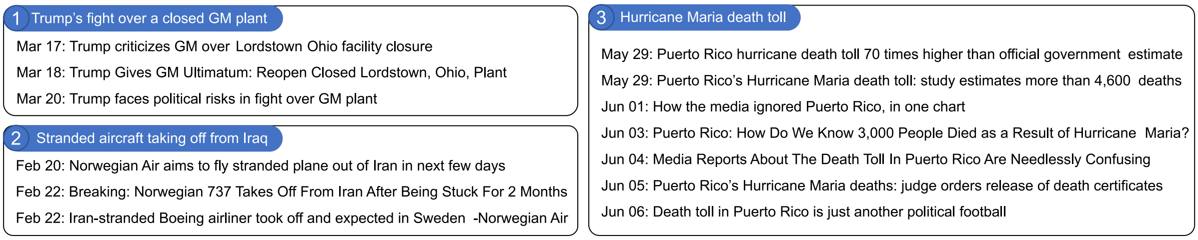
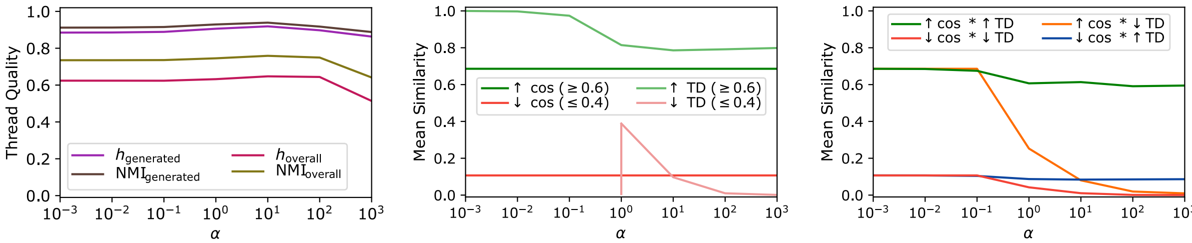


Fig. 7. Sample threads identified by our 5W1H-HAC approach (TD-miniLM config.) from the NewSHead collection.



(a) Thread Quality for different values of Time Decay factor α.

(b) Mean Cosine (cos) and Time Decay (TD) similarity.

(c) Effect of TD similarity on overall $cos * $ TD similarity.

Fig. 8. Impact of Time Decay (TD) factor α on the thread quality and overall similarity score in HAC. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Comparison of HAC and K-means for news topic clustering.

Method	Run time		Cluster quality	
	Total	Average (per split)	h	NMI
K-Means	**54.56 s**	**18.19 s**	**0.4404**	**0.4179**
HAC	8 m 53.46 s	2 m 57.82 s	0.3670	0.3677

is the normalised time decay between the original documents (c.f. Section 3.2). Fig. 8(b) shows the mean cos and TD similarity scores of the document-pairs that have high (≥ 0.6) and low (≤ 0.4) similarity scores, respectively. Fig. 8(c) presents the $cos *$ TD similarity scores of the document pairs that have high and low similarity scores based on cos and TD, respectively, i.e., overall four groups of document pairs that have either (1) high cos and TD, (2) low cos and TD, (3) high cos but low TD, or (4) low cos but high TD. For the document-pairs with low cos and high TD (in blue), we observe that the TD component does not increase the overall similarity score even for higher values of α. This is an essential property showing that the inclusion of the TD component does not favour documents with a small time gap if the content similarity between the documents is low. Most importantly, for the document-pairs with high cos but low TD (in orange), for $\alpha > 0.1$, the document-pairs with high cos have low $cos *$ TD similarity scores. Therefore, from Figs. 8(a) and 8(c), we conclude that the improvements in thread quality are related to the variations in the similarity scores caused by the time decay factor α. Moreover, the decline in thread quality for higher values of α (> 100) is related to the penalisation of the document-pairs with low TD scores as the $cos *$ TD score tends to 0.

Overall, the best values for α in this case are observed at $0.1 < \alpha \leq 100$ (i.e., $\alpha \in \{1, 10, 100\}$). This is an important analysis to select the right time decay factor in unsupervised tasks such as information threading.

7.3. Efficiency of HAC for information threading compared to K-means clustering

As briefly discussed in Section 3.3, due to the large number of clusters in the information threading task, we argue that HAC is a more suitable clustering technique compared to more popular techniques such as K-Means. In this analysis, we compare the efficiency of HAC compared to K-Means for the information threading task and contrast it with a general news topic clustering task. All evaluations in this analysis were performed on an Intel(R) Xeon(R) Gold 6244 CPU @ 3.60 GHz with 64 GB memory, and the time taken is reported as an average of 10 runs.

In general clustering tasks where the number of clusters is usually much smaller than the number of items to be clustered, K-Means is considered as a more efficient algorithm compared to HAC (Shetty & Singh, 2021; Singh & Singh, 2012). We perform news topic clustering on the NewSHead collection based on the 8 topic clusters presented by Gu et al. (2020). Similar to our information threading experiments, we perform news topic clustering separately on the 3 test sets of the NewSHead collection (c.f. Section 4.1). Table 5 presents the run time of K-Means and HAC along with the quality of the identified clusters in terms of homogeneity (h) and

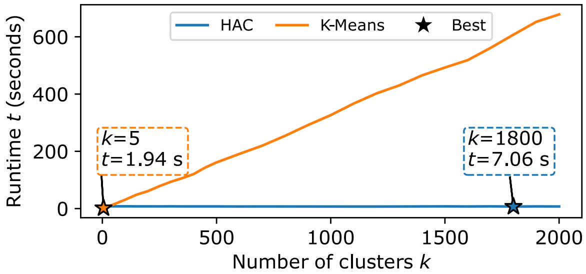


Fig. 9. Effect of increasing number of clusters on the time taken by the clustering algorithm.

Table 6
Comparison of thread quality and time taken by K-Means clustering and HAC for information threading.

Configurations	Run time		Generated threads		Overall performance	
	Total	Average (per split)	h	NMI	h	NMI
K-Means	9 h 21 m 33.89 s	3h 07 m 11.30 s	0.6458	0.7848	0.0001	0.0003
5W1H-KMeans	101 h 12 m 25.67 s	33h 44 m 08.56 s	0.8833	0.9049	**0.6476**	**0.7539**
5W1H-HAC-W	16 m 06.60 s	05 m 22.20 s	0.8947	0.9129	0.5937	0.7157
5W1H-HAC-TD	**09 m 14.02 s**	**03 m 04.67 s**	**0.9144**	**0.9348**	0.6329	0.7537

NMI. From Table 5 we observe that K-Means is more efficient (−89.77% total run time) and more effective (+20.00% homogeneity and +13.64% NMI) compared to HAC for topic clustering.

However, compared to HAC, the efficiency of K-Means is negatively affected when the number of clusters is increased. We show this by performing K-Means clustering on randomly initialised isotropic Gaussian blobs with different numbers of centres (k, i.e., clusters). In particular, we generated isotropic Gaussian blobs with 10,000 data points, and k in the range $[5, 2000]$ with unit standard deviation of the clusters. Fig. 9 shows the effect of increasing the number of clusters on the time taken by K-Means and HAC. From Fig. 9, we observe that the time taken by K-Means increases linearly with the number of clusters. In contrast, the time taken by HAC remains comparable across different values of k. Moreover, the lowest time taken by HAC is for $k = 1800$, which shows that HAC is more efficient compared to K-Means for a higher number of clusters.

To further investigate the efficiency of HAC and K-Means for the information threading task, we evaluate our proposed threading approach by replacing HAC with K-Means clustering. Table 6 presents the time taken by our proposed information threading approach with K-Means and HAC clustering (5W1H-KMeans and 5W1H-HAC). We deploy all the 5W1H configurations with the miniLM representations of the pseudo-passages. Table 6 also shows the time taken by the K-Means document clustering baseline that we described in Section 4.2. From Table 6 we observe that HAC based information threading is markedly more efficient than K-Means based threading (e.g. −99.85% total run time by 5W1H-HAC-TD compared to 5W1H-KMeans). In addition, the quality of 5W1H-KMeans threads is comparable to 5W1H-HAC threads, where 5W1H-HAC slightly outperforms 5W1H-KMeans under the Generated Threads setup and slightly underperforms under the Overall Performance setup. Moreover, the proposed 5W1H-HAC approach is markedly more efficient compared to the K-Means document clustering baseline (−98.36% total run time).

Overall from this analysis, we conclude that, although both HAC and K-Means can be effective for information threading, HAC is a much more efficient clustering technique for the information threading task compared to K-Means clustering. In particular, HAC's bottom-up algorithm is well-suited for the information threading task, where the number of clusters is much higher than the general topic-based clustering task. Moreover, our proposed configuration for the deployment of HAC for information threading (i.e., 5W1H-HAC-TD based on complete linkage and TD similarity, c.f. Section 3) is the most effective and efficient (c.f. Table 6).

8. Discussion

In this section, we first discuss our observations from the results of our experiments in Section 8.1. We further discuss the theoretical and practical implications of our work in Section 8.2.

8.1. Comparison with existing work

We now discuss our observations comparing the findings from our offline evaluation of thread quality (c.f. Section 5), and our user study of human preferences (c.f. Section 6.5). The offline evaluation showed that our 5W1H-HAC approach can markedly improve the number of identified threads in a collection (Figs. 4(c) and 4(d)), while maintaining the quality of the identified threads, as measured by Homogeneity and NMI (Table 2). Moreover, our user study showed that the threads from 5W1H-HAC are preferred by the users and are rated highest in terms of coherence, diversity and chronological correctness.

As discussed in Section 6.5, EventX was found to be the least preferred approach in our user study. This is consistent with the quality scores of the threads that we presented under the Generated Thread setup in Section 5.2. However, from Figs. 4(c) and

4(d), we observe that EventX generated more threads than k-SDPP, which improves its Overall Performance compared to k-SDPP. In Section 6.5, we also found that EventX is preferred to k-SDPP in terms of coherence, but is less preferred to k-SDPP in terms of diversity and chronological correctness. This variation in preferences for EventX is likely due to EventX being based on event clustering, which requires the documents in a cluster to discuss a particular event, irrespective of whether the documents provide repeated or diverse information about the event. Overall, we conclude that, when comparing k-SDPP and EventX, event clustering (EventX) is potentially effective in identifying a higher number of events in a collection. However, event clustering is not well suited for information threading, since unlike k-SDPP's or 5W1H-HAC's threads, EventX's event clusters do not necessarily account for diversity of information about the same event.

In terms of information diversity, as discussed in Section 6.5, we found that the k-SDPP threads are comparable to the threads from 5W1H-HAC (Fig. 6(b)). However, as shown in Fig. 4(a), k-SDPP identifies the least number of threads compared to EventX or 5W1H-HAC. This is expected since k-SDPP aims to identify threads about the most important events in a collection, to enable users to understand the collection as a whole. Therefore, k-SDPP is not entirely effective for information threading, since information threading aims to identify all possible threads in a collection to enable various information seekers to quickly find related information.

8.2. Theoretical and practical implications

From the theoretical point of view, previous related studies are primarily focused on identifying threads of specific documents (e.g. Gillenwater et al., 2012; Shahaf & Guestrin, 2012) or event-based clusters of documents (e.g. Liu et al., 2020; Nallapati et al., 2004) in a collection. Differently, this paper presents a generalised concept of threading in a document collection to identify coherent and chronological sequences of documents that describe a particular event, activity or discussion. In addition, the findings from our experiments show that 5W1H questions are important features for the effective identification of information threads that describe specific events in a document collection. Therefore, this work promotes a new research direction for future work in leveraging 5W1H questions for effective *information threading* using further sophisticated techniques, for example, network analysis of the extracted 5W1H questions' answers from a document collection.

Our work also has practical implications for human users to facilitate the understanding of specific events' chronology from multiple documents. In particular, information threads generated by our proposed approach can help human users to quickly make sense of coherent information about an event from a massive pool of digital documents such as online news articles. Moreover, with the focus on identifying a maximum number of threads in a collection, our proposed approach can contribute to providing a threaded structure to unstructured document collections. In addition, we present important discussions for estimating the best parameters of our proposed approach without the ground truth thread labels, i.e., parameters to determine coherence and information diversity for candidate thread selection (c.f. Section 3.4) and the time-decay factor (c.f. Section 7.2). These discussions help to enable the practical application of our work on real-world datasets, where there is no prior-knowledge available about the actual threads.

9. Conclusions

We have proposed a novel unsupervised information threading approach named 5W1H-HAC. Our approach generates coherent information threads by leveraging hierarchical agglomerative clustering (HAC) based on the temporal relationships and answers to journalistic 5W1H questions from documents. The information threads that are produced by our approach can help information seekers to quickly find time-ordered and diverse information about an event, activity or discussion within a large unstructured collection of documents. Using the NewSHead and Multi-News test collections, we conducted an offline evaluation to evaluate the quality of the threads that are generated by our 5W1H-HAC approach. We also conducted a user study to evaluate the preferences of human users in terms of coherence, information diversity, chronological correctness and overall preference for the generated threads. Our offline evaluation showed that our 5W1H-HAC approach markedly outperforms the K-Means document clustering, the k-SDPP document threading and the EventX event extraction approaches from the literature, in terms of thread quality. Moreover, our user study showed that a significant (chi-square goodness-of-fit test, $p < 0.05$) proportion of the study participants, (i.e., users), preferred the threads that are generated by 5W1H-HAC compared to the threads from k-SDPP and EventX. Furthermore, the user study participants rated the threads from 5W1H-HAC significantly (paired samples t-test, $p < 0.05$) higher in terms on coherence, information diversity, chronological correctness.

As future work, we plan to investigate hierarchical associations between documents in information threads that can provide a hierarchical threaded structure to unstructured collections. We also plan to analyse the underlying factors that affect the coherence of information threads, to develop a more succinct evaluation metric for evaluating the effectiveness of information threading approaches.

CRediT authorship contribution statement

Hitarth Narvala: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Project administration, Writing – original draft. **Graham McDonald:** Supervision, Conceptualization, Methodology, Investigation, Project administration, Writing – review & editing. **Iadh Ounis:** Supervision, Conceptualization, Methodology, Investigation, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets used for evaluating the proposed approach are publicly available (i.e., NewSHead and Multi-News).

References

Aggarwal, C. C., & Subbian, K. (2012). Event detection in social streams. In *Proceedings of the 2012 SIAM international conference on data mining* (pp. 624–635). http://dx.doi.org/10.1137/1.9781611972825.54.

Allan, J. (2012). *Topic detection and tracking: Event-based information organization. volume 12*. Springer US, http://dx.doi.org/10.1007/978-1-4615-0933-2.

Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study final report. In *Proceedings of the DARPA broadcast news transcription and understanding workshop*. http://dx.doi.org/10.1184/R1/6626252.V1.

Cai, D., He, X., & Han, J. (2005). Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 1624–1637. http://dx.doi.org/10.1109/TKDE.2005.198.

Chen, C. C., & Wang, H. C. (2021). Adapting the influences of publishers to perform news event detection. *Journal of Information Science*, http://dx.doi.org/10.1177/01655515211047422.

Churchill, R., & Singh, L. (2022). The evolution of topic modeling. *ACM Computing Surveys*, *54*, http://dx.doi.org/10.1145/3507900.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197. http://dx.doi.org/10.1109/4235.996017.

Fabbri, A., Li, I., She, T., Li, S., & Radev, D. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics* (pp. 1074–1084). http://dx.doi.org/10.18653/v1/P19-1102.

Fan, W., Guo, Z., Bouguila, N., & Hou, W. (2021). 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*.

Gillenwater, J., Kulesza, A., & Taskar, B. (2012). 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* (pp. 710–720).

Gu, X., Mao, Y., Han, J., Liu, J., Wu, Y., Yu, C., et al. (2020). Generating representative headlines for news stories. In *Proceedings of the web conference* (pp. 1773–1784). http://dx.doi.org/10.1145/3366423.3380247.

Hamborg, F., Breitinger, C., & Gipp, B. (2019). 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*.

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C. R., Han, J., et al. (2016). Liberal event extraction and event schema induction. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (Volume 1: Long papers)* (pp. 258–268). http://dx.doi.org/10.18653/v1/P16-1025.

Jacobs, G., & Hoste, V. (2020). Extracting fine-grained economic events from business news. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 235–245).

Kulesza, A., & Taskar, B. (2010). Structured determinantal point processes. In *Proceedings of the advances in neural information processing systems*.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*, 79–86.

Kuo, J. J., & Chen, H. H. (2007). Cross-document event clustering using knowledge mining from co-reference chains. *Information Processing & Management*, *43*, 327–343. http://dx.doi.org/10.1016/J.IPM.2006.07.016.

Lee, C., Lee, G. G., & Jang, M. (2007). Dependency structure language model for topic detection and tracking. *Information Processing & Management*, *43*, 1249–1259. http://dx.doi.org/10.1016/j.ipm.2006.02.007.

Liu, B., Han, F. X., Niu, D., Kong, L., Lai, K., & Xu, Y. (2020). Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data*, *14*, http://dx.doi.org/10.1145/3377939.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*, 129–137. http://dx.doi.org/10.1109/TIT.1982.1056489.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Mele, I., Bahrainian, S. A., & Crestani, F. (2019). Event mining and timeliness analysis from heterogeneous news streams. *Information Processing & Management*, *56*, 969–993. http://dx.doi.org/10.1016/j.ipm.2019.02.003.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, *26*, 354–359.

Nallapati, R., Feng, A., Peng, F., & Allan, J. (2004). Event threading within news topics. In *Proceedings of the 13th ACM international conference on information and knowledge management* (pp. 446–453). http://dx.doi.org/10.1145/1031171.1031258.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Qian, Y., Deng, X., Ye, Q., Ma, B., & Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management*, *56*, Article 102086. http://dx.doi.org/10.1016/J.IPM.2019.102086.

Reimers, N., & Gurevych, I. (2019). 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* (pp. 3982–3992). http://dx.doi.org/10.18653/v1/D19-1410.

Röder, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the 8th ACM international conference on web search and data mining* (pp. 399–408). http://dx.doi.org/10.1145/2684822.2685324.

Rosenberg, A., & Hirschberg, J. (2007). 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*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the 5th workshop on energy efficient machine learning and cognitive computing - NeurIPS edition*.

Saravanakumar, K. K., Ballesteros, M., Chandrasekaran, M. K., & McKeown, K. (2021). 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* (pp. 2330–2340). http://dx.doi.org/10.18653/v1/2021.eacl-main.198.

Shahaf, D., & Guestrin, C. (2012). Connecting two (or less) dots: Discovering structure in news articles. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *5*, http://dx.doi.org/10.1145/2086737.2086744.

Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., & Leskovec, J. (2013). Information cartography: creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1097–1105). http://dx.doi.org/10.1145/2487575.2487690.

Shetty, P., & Singh, S. (2021). Hierarchical clustering: a survey. *International Journal of Applied Research*, *7*, 178–181. http://dx.doi.org/10.22271/allresearch.2021.v7.i4c.8484.

Singh, N., & Singh, D. (2012). Performance evaluation of k-means and heirarichal clustering in terms of accuracy and running time. *International Journal of Computer Science and Information Technologies*, *3*, 4119–4121.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems* (pp. 6000–6010).

Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244. http://dx.doi.org/10.1080/01621459.1963.10500845.

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd international conference on international conference on machine learning* (pp. 478–487).

Yu, H., Zhang, Y., Ting, L., & Sheng, L. (2007). Topic detection and tracking review. *Journal of Chinese Information Processing*, *6*, 77–79.

Zhang, Y., Guo, F., Shen, J., & Han, J. (2022). Unsupervised key event detection from massive text corpora. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2535–2544). http://dx.doi.org/10.1145/3534678.3539395.

Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., & Buntine, W. (2021). Topic modelling meets deep neural networks: A survey. In *Proceedings of the thirtieth international joint conference on artificial intelligence* (pp. 4713–4720). http://dx.doi.org/10.24963/ijcai.2021/638.

Zong, C., Xia, R., & Zhang, J. (2021). Topic detection and tracking. (pp. 201–225). http://dx.doi.org/10.1007/978-981-16-0100-2_9.