Online Visual Analytics of Text Streams

Shixia Liu, Jialun Yin, Xiting Wang, Weiwei Cui, Kelei Cao, Jian Pei

Abstract—We present an online visual analytics approach to helping users explore and understand hierarchical topic evolution in high-volume text streams. The key idea behind this approach is to identify representative topics in incoming documents and align them with the existing representative topics that they immediately follow (in time). To this end, we learn a set of streaming tree cuts from topic trees based on user-selected focus nodes. A dynamic Bayesian network model has been developed to derive the tree cuts in the incoming topic trees to balance the fitness of each tree cut and the smoothness between adjacent tree cuts. By connecting the corresponding topics at different times, we are able to provide an overview of the evolving hierarchical topics. A sedimentation-based visualization has been designed to enable the interactive analysis of streaming text data from global patterns to local details. We evaluated our method on real-world datasets and the results are generally favorable.

Index Terms—streaming text data, evolutionary tree clustering, streaming tree cut, streaming topic visualization.

✦

1 INTRODUCTION

Surveying and exploring text streams that have many hierarchical and evolving topics are important aspects of many big data applications [9], [36]. For example, the use of such evolving hierarchical topics allows for the detection and tracking of new and emerging events (e.g., Ebola outbreak) in a huge volume of streaming news articles and microblog posts. Exciting progress, such as learning topics from text streams, has been made in mining text streams [36]. However, one essential problem remains: how can we effectively present interesting topics and track their evolution over time in a comprehensible and manageable manner? This task is a key to connecting big data with people.

Let us consider an example to understand this challenge. Suppose an analyst reads an article entitled "Third U.S. Aid Worker Infected with Ebola Arrives in Nebraska." The analyst is interested in the topic "Ebola-infected aid workers" and wants to analyze the relevant discussions in the subsequent weekly news articles. In addition, s/he is interested in how this topic is related to other topics in the news stream as time progresses, especially the newly generated topics. Such analysis helps the analyst understand the relationship between the severity of Ebola and the intensity of public opinion. Based on this relationship, s/he can make suggestions to the government.

A text stream, such as the aforementioned Ebola dataset, often contains hundreds or even thousands of topics that can be naturally organized in a tree, known as a topic tree [7], [36], [44]. A topic tree may change as new documents arrive. We can mine a sequence of coherent topic trees to represent major topics in the text stream and their evolution over time [36]. However, the question of whether such a sequence of topic trees is effective enough to analyze and understand a text stream remains, in particular, whether these topic trees can illustrate the accumulation and aggregation of the new documents into the existing topics.

To address this problem, we have developed a visual analytics system, *TopicStream*, to help users explore and understand hierarchical topic evolution in a text stream. Specifically, we incrementally extract a new tree cut from the incoming topic tree, based on a dynamic Bayesian network (DBN) model. We model the topics that a user is interested in as proper tree cuts in a sequence of topic trees similar to [12]. A tree cut is a set of tree nodes describing the layer of topics that a user is interested in. In *TopicStream*, we employ the DBN model to derive the tree cut from an incoming topic tree. A time-based visualization is then developed to present the hierarchical clustering results and their alignment over time. In particular, we have adopted a customized sedimentation metaphor to visually illustrate how incoming text documents are aggregated over time into the existing document archive, including document entrance into the scene from an entrance point, suspension while approaching to the topic, accumulation and decay on the topic, as well as aggradation with the topic over time [36].

We make the following technical contributions in this work:

- **A streaming tree cut algorithm** is proposed to extract an optimal tree cut for an incoming topic tree based on user interests. This algorithm produces a sequence of representative topic sets for different topic trees, which smoothly evolve over time.
- **A sedimentation-based metaphor** is integrated into the river flow metaphor to visually illustrate how new documents are aggregated into old documents. It helps analysts immediately track and understand incoming topics and connect those topics with existing ones.
- **A visual analytics system** is built to integrate evolutionary hierarchical clustering [36] and the streaming tree cut techniques into an interactive visualization. The

- *S. Liu is with School of Software, Tsinghua University.*
  E-mail: shixia@tsinghua.edu.cn.
- *Jialun Yin, Xiting Wang, and Kelei Cao are with Tsinghua University.*
  E-mail: {yinjl14, wang-xt11, ckl13}@mails.tsinghua.edu.cn.
- *Weiwei Cui is Microsoft Research. E-mail: weiwei.cui@microsoft.com.*
- *Jian Pei is with Simon Fraser University, Burnaby, BC Canada.*
  E-mail: jpei@cs.sfu.ca.

unique feature of this system is its ability to provide a coherent view of evolving topics in text streams.

2 RELATED WORK

2.1 Evolutionary Topic Analysis

Various generative-probabilistic-model-based machine learning algorithms, such as dynamic Latent Dirichlet Allocation (LDA) [6] and hierarchical Dirichlet processes [4], [5], [43], [45], have been proposed to extract evolving topics from a text stream. MemeTracker [23] was developed to effectively identify phrase-based topics from millions of news articles. In many applications, evolving topics may be related to or correspond to one another over time. The most intuitive relationships are *topic correlation* [37] and *common topics* [38]. Recent efforts have focused on the analysis of topic evolution patterns in text data, including topic birth, death, splitting, and merging [18]. Although the aforementioned methods help users understand a text corpus, none of them focus on mining and understanding streaming hierarchical topics.

Some efforts have also been exerted recently to mine hierarchical topics and their evolving patterns in temporal datasets. The evolutionary hierarchical clustering algorithm [9] generates a sequence of hierarchical clusters. The major feature of this algorithm is that clustering properly fits current data at any time (fitness). Furthermore, clustering does not shift dramatically from one time step to the next when content is similar (smoothness). However, this algorithm can only generate evolving binary trees. To tackle this issue, Wang et al. [36] formulated the multi-branch tree construction problem as a Bayesian online filtering process. Unlike the method proposed in [36], our method addresses the problem of better understanding and analyzing a sequence of evolutionary multi-branch topic trees. We first learn a set of evolutionary tree cuts from the topic trees based on the user-selected focus nodes. Then we design a sedimentation-based interactive visualization to reveal hierarchical topic evolution from multiple perspectives.

2.2 Visual Topic and Event Evolution

The visual analysis of evolving topics in text corpora has been widely studied in recent years [11], [19], [24], [32]. Many methods utilize a river metaphor (a stacked graph) to convey evolving topics over time. For example, ThemeRiver [19] visually depicts how keyword strengths change over time in a text corpus through a river metaphor. A layer represents a topic in this metaphor. The varying width of a layer represents strength change over time. TIARA [25], [26] employs an enhanced stacked graph to illustrate how topics evolve over time. ParallelTopics [14] utilizes ThemeRiver to illustrate topic evolution over time and parallel coordinate plots to reveal the probabilistic distribution of a document on different topics. TextFlow [11] was developed to help analysts visually analyze topic merging and splitting relationships and track their evolution over time. A visual analysis system was designed by Xu et al. [42] to allow analysts to interactively explore and understand the dynamic competition relationships among topics. Recently, Sun et al. [33] extended this work to study both the cooperation and competition relationships among topics.

Several visualization techniques have been proposed recently to help users analyze temporal events and their evolving patterns [15], [22], [28]. EventRiver [28] assumes that clusters of news articles with similar content are adjacent in time and can be mapped to events. Thus, this method automatically detects important events and visually presents their impact over time. LifeFlow [40] and Outflow [39] help users explore temporal event sequences.

The aforementioned approaches focus on the visual exploration of evolving topics/events with flat structures. By contrast, our method attempts to support the visual analysis of evolving hierarchical topics over time.

HierarchicalTopics [16] hierarchically organizes topics using the BRT model [7], [27], which can then represent a large number of topics without being cluttered. However, this method utilizes one static tree to organize all topics and cannot illustrate splitting/merging relationships among topics.

To solve this problem, Cui et al. [12] developed *RoseRiver* to progressively explore and analyze the complex evolution patterns of hierarchical topics. This system introduces the concept of evolutionary tree cuts to help better understand large document collection with time-stamps. However, it fails to provide a mechanism to analyze streaming data because a global tree cut algorithm is employed. In addition, the authors used the DOI-based heuristic rule to derive the key tree cut, which may not be the optimal solution.

Unlike the preceding method, we employ a-posterior-probability-based method to estimate the fitness of the tree cut. We then formulate the derivation of the tree cut in incoming data as a DBN. The quantitative evaluation in Sec. 6 shows that the posterior-probability-based method performs better than the DOI-based method in [12] to fit the focus nodes and topic trees. The performance of the DBN-based streaming tree cut algorithm is comparable with that of the global tree cut algorithm proposed in [12]. These observations demonstrate the effectiveness of the proposed mining algorithms at handling incoming data in text streams. An improved visual sedimentation metaphor [21] has been adopted to visually illustrate how incoming text streams aggregate into existing topics.

3 SYSTEM OVERVIEW

TopicStream is designed to track and understand the dynamic characteristics of text streams. It consists of two major modules: streaming tree cut and streaming visualization (Fig. 1).

The input of the streaming tree cut is a set of topic trees with tree cuts and a set of incoming documents. In *TopicStream*, the topic trees are derived by the evolutionary tree clustering method developed by Wang et al. [36]. The basic idea of this method is to balance the fitness of a tree and the smoothness between trees by a Bayesian online filtering process. We derive the tree cuts based on the user-selected focus node(s). This module initially extracts a topic tree from the
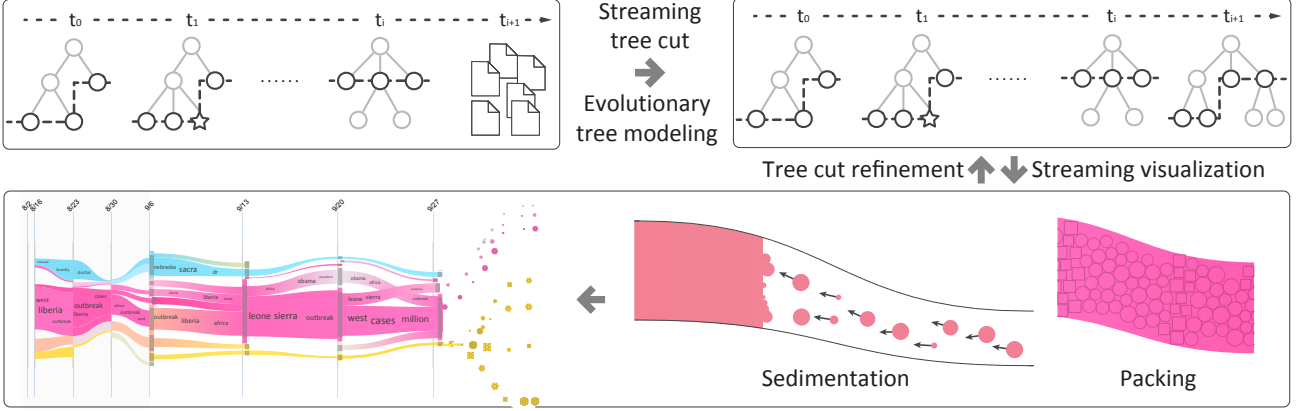
Fig. 1: *TopicStream* system consists of two modules: streaming tree cut and streaming visualization.
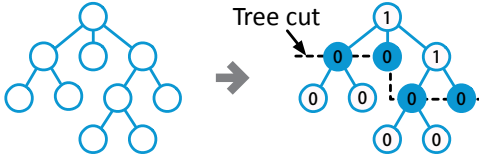

Fig. 2: Tree cut example: the cut is denoted by the dotted line. Every node above the dotted line is labeled 1, while the others are labeled 0.

newly arrived documents using the evolutionary tree clustering model [36]. A tree cut is then derived from the new topic tree through the developed streaming tree cut algorithm.

The streaming tree cuts are then fed into the visualization module. We employ the visual sedimentation metaphor to reveal the merging process of newly arrived documents with the dominant center of visualization. The circle packing algorithm is also developed to illustrate the relationships of document clusters within each topic stripe, including their similarity and temporal relationships [35], [46].

4 STREAMING TREE CUT

This section explains how a new tree cut is incrementally derived as new text data arrives.

4.1 Problem Formulation

We use a tree cut to represent a topic tree based on user interests, which is similar to [12]. A **tree cut** is a set of nodes in which every path from the root of the tree to a leaf contains exactly one node from the cut. Thus, each cut can be used as a set of representative topic nodes. That is, a tree cut represents a level of topic granularity of a user's interests. Fig. 2 represents an example of a tree cut. We refer to each node on the tree cut as a "**cut node**."

The basic principle to determine a set of optimal tree cuts is that each tree cut in the sequence should be similar to the one at the previous time step if the tree structures are similar (smoothness). The tree cut must also adequately represent user interests and the topic tree at that time step (fitness). The global tree cut algorithm developed in [12] computes all tree cuts simultaneously based on the focused nodes. Two problems arise when applying the aforementioned algorithm
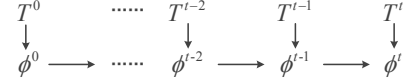

Fig. 3: Dynamic Bayesian network for deriving streaming tree cuts. Here T^t and ϕ^t are the topic tree and tree cut at t.

to the text stream. First, it is very time consuming to compute all the tree cuts each time new text data arrives. Second, if the tree cuts are recomputed along with the new data, then the existing tree cuts are changed to a certain extent, which makes maintaining the mental map of analysts difficult [41].

To solve the aforementioned problems, we have adopted a dynamic Bayesian network (DBN) model to infer the tree cut for the incoming text data organized by a topic tree. Previous studies have shown that adopting overlapping successive views to support continuity across data sets is a frequently adopted principle to process temporal data [9], [41]. In our case, the new tree cut ϕ^t is relevant to temporally adjacent tree cuts as well as T^t (Fig. 3). In particular, topic mapping between adjacent trees is utilized as a constraint to smooth tree cut transitions over time.

4.2 Model

Assume that we already have a sequence of topic trees and the corresponding tree cuts. The problem of deriving a new tree cut in a text stream can then be regarded as a labeling problem. The topic nodes above the tree cut are labeled

Notation	Definition
T^t	The topic tree at time t
ϕ^t	The tree cut at time t
m	Number of focus nodes selected by the user
T_{fi}	The ith focus node
$\mathscr{D}_{fi}$	The document set of the ith focus node
$p(\phi^t|\phi^{t-1}, T^t)$	The conditional distribution of ϕ^t according to DBN
$p(\phi^t|T^t)$	Fitness of tree cut ϕ^t to T^t
$p(\phi^t|\phi^{t-1})$	Smoothness between adjacent tree cuts ϕ^t and ϕ^{t-1}
$p(\phi^t|\mathscr{D}_{f0}, ..., \mathscr{D}_{fm})$	The posterior probability of a tree cut ϕ^t
$E_1(T^t)$	The similarity energy of the topic tree T^t
T_r, T_s	A topic node (an internal node) in the topic tree
$\mathbf{S}(T_r, T_s)$	The cosine similarity between topic nodes T_r and T_s
l_s	The label (0 or 1, see Fig. 2) of topic node T_s
$E_2(\phi^t, \phi^{t-1})$	The smoothness energy between tree cuts ϕ^t and ϕ^{t-1}
$\mathscr{D}_s$	The document set of topic node T_s
$f_{DCM}(\mathscr{D})$	The marginal distribution of document set $\mathscr{D}$
$\mathbf{WS}(T_c)$	Window size for T_c in mean-shift clustering

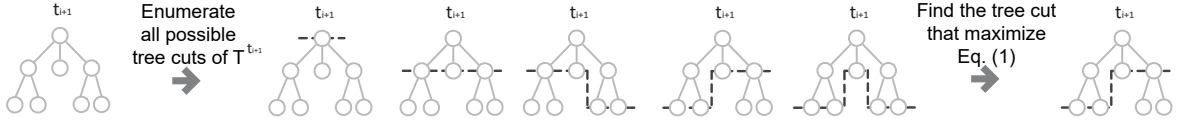TABLE 1: Frequently used notations in the model.

Fig. 4: Streaming tree cut algorithm: given the incoming topic tree $T^{t_{i+1}}$, we enumerate all possible tree cuts of $T^{t_{i+1}}$ and then pick the tree cut that maximizes Eq. (1).

1, whereas the rest (including the cut nodes) are labeled 0 (Fig. 2). We first introduce some frequently used notations in Table 1, which are useful for subsequent discussions.

Given m focus nodes $\{T_{fi}\}$ with document sets $\{\mathscr{D}_{fi}\}$, we infer the tree cut ϕ^t in the incoming topic tree T^t. Fig. 3 shows that T^t is an observation variable and ϕ^t is a hidden variable. The relationship between ϕ^t and ϕ^{t-1}, as well as T^t, can be modeled by DBN. Accordingly, the conditional distribution of ϕ^t is $p(\phi^t|\phi^{t-1},T^t)$. Since ϕ^t is relevant to $\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}$ at each time t, we formulate the inference of the new tree cut as:

$$max \ p(\phi^t,\phi^{t-1},...,\phi^0|\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}) \cdot p(\phi^t|\phi^{t-1},T^t). \quad (1)$$

As shown in Fig. 4, the goal is to find the tree cut that maximizes Eq. (1).

Since $\phi^t,\phi^{t-1},...,\phi^0$ are conditionally independent given $\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}$, the first term is computed by $\prod_{\tau=0}^{t} p(\phi^\tau|\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm})$. According to the graphical model of DBN (Fig. 3), the second term is proportional to $p(\phi^t|T^t)p(\phi^t|\phi^{t-1})$. Because ϕ^{t-1}, $\phi^{t-2},...$, ϕ^0 are known, Eq. (1) can be simplified as:

$$max \ p(\phi^t|T^t)p(\phi^t|\phi^{t-1})p(\phi^t|\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}). \quad (2)$$

$p(\phi^t|T^t)$ denotes how well the tree cut ϕ^t represents T^t, which is measured by the similarity energy $\mathbf{E}_1$ in the form of $p(\phi^t|T^t) = e^{-E_1(T^t)}$. $\mathbf{E}_1$ measures the content similarity of each topic T_r in T^t for the two topic sets, which are topic node sets labeled 0 and 1, respectively.

$$E_1(T^t) = \sum_{T_r \in \mathscr{N}^t} \min_{T_s \in \mathscr{N}^t, l_s = l_r} (-\log(\mathbf{S}(T_r,T_s))), \quad (3)$$

where l_r is the label (1 or 0) of topic node T_r, $\mathscr{N}^t$ is the set which contains all tree nodes in T^t. For a topic T_r, the cosine similarity $\mathbf{S}(T_r,T_s)$ is used to compute the similarity value between T_r and T_s with the same label.

$p(\phi^t|\phi^{t-1})$ measures the smoothness cost between two adjacent tree cuts using the smoothness energy $\mathbf{E}_2$, which is defined as $p(\phi^t|\phi^{t-1}) = e^{-E_2(\phi^t,\phi^{t-1})}$. $\mathbf{E}_2$ measures the mapping similarity between T^t and T^{t-1}:

$$E_2(\phi^t,\phi^{t-1}) = \sum_{T_r \in \mathscr{N}^t, T_s \in \mathscr{N}^{t-1}} |l_r - l_s|\varphi(l_r,l_s), \quad (4)$$

where $\varphi(l_r,l_s)$ denotes the mapping weight computed by the evolutionary tree clustering model.

$p(\phi^t|\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm})$ is defined as a posterior probability of a tree cut ϕ^t. Thus,

$$p(\phi^t|\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}) \propto p(\mathscr{D}_{f1},\mathscr{D}_{f2},...,\mathscr{D}_{fm}|\phi^t)p(\phi^t), \quad (5)$$

where $p(\mathscr{D}_{f0},\mathscr{D}_{f1},...,\mathscr{D}_{fm}|\phi^t)$ is the likelihood of the tree cut. $p(\phi^t)$ indicates the prior preference of the node number for the tree cut. The tree cut that results in maximum posterior probability is the optimal tree cut.

$p(\phi)$ is defined as $e^{-\lambda|\mathscr{C}_\phi|}$, where $\mathscr{C}_\phi$ is the set of topics in ϕ,

$|\mathscr{C}_\phi|$ is the node number in the tree cut, and λ is the parameter that balances the likelihood and expected node number.

We then illustrate how the likelihood of a tree cut can be calculated. We adopt a prediction model to estimate the likelihood of each possible tree cut. For simplicity's sake, we begin with one focus node. Given a focus node T_f and its corresponding document set $\mathscr{D}_f$, the predictive probability of a tree cut ϕ is defined as:

$$p(\mathscr{D}_f|\phi) = \sum_{T_s \in \mathscr{C}_\phi} \omega_s p(\mathscr{D}_f|\mathscr{D}_s), \quad (6)$$

where $\mathscr{C}_\phi$ is the set of topics in ϕ and ω_s is the prior probability that all the documents in $\mathscr{D}_f$ belong to $\mathscr{D}_s$. To calculate ω_s, we assume that the probability of a set of documents belonging to a specific topic is proportional to the number of documents in that topic [8]. Accordingly, we obtain $\omega_s = |\mathscr{D}_s|/|\mathscr{D}_a|$. $\mathscr{D}_a$ includes all documents in a tree. $p(\mathscr{D}_f|\mathscr{D}_s)$ is the predictive distribution of the corresponding topic model.

$$p(\mathscr{D}_f|\mathscr{D}_s) = f(\mathscr{D}_f \cup \mathscr{D}_s)/f(\mathscr{D}_s), \quad (7)$$

where $f(\mathscr{D})$ is the marginal probability of data $\mathscr{D}$.

The Dirichlet compound multinomial (DCM) distribution is derived from multinomial and Dirichlet conjugate distributions [27]. Because it relies on hierarchical Bayesian modeling techniques, DCM is a more appropriate generative model than the traditional multinomial distribution for text documents. Thus, we utilize the DCM distribution to represent the marginal distribution $f(\mathscr{D})$ as follows:

$$f_{DCM}(\mathscr{D}) = \prod_{i}^{n} \frac{(\sum_{j}^{|\mathscr{V}|} z_i^{(j)})!}{\prod_{j}^{|\mathscr{V}|} z_i^{(j)}!} \cdot \frac{\Delta(\boldsymbol{\alpha} + \sum_i \mathbf{z}_i)}{\Delta(\boldsymbol{\alpha})}, \quad (8)$$

where $|\mathscr{V}|$ is the vocabulary size, $\mathbf{z}_i \in \mathscr{R}^{|\mathscr{V}|}$ is the word vector of the ith document, and $z_i^{(j)}$ is the frequency of the jth term. $\boldsymbol{\alpha} = (\alpha^{(1)},\alpha^{(2)},...,\alpha^{(|\mathscr{V}|)})^T \in \mathscr{R}^{|\mathscr{V}|}$ is the parameter that controls the Dirichlet distribution, which is the prior of the multinomial distribution of each topic. $\Delta(\boldsymbol{\alpha})$ is the Dirichlet delta function defined by $\Delta(\boldsymbol{\alpha}) = \Gamma(\sum_{j=1}^{|\mathscr{V}|} \alpha^{(j)})/\prod_{j=1}^{|\mathscr{V}|} \Gamma(\alpha^{(j)})$. The gamma function has the property $\Gamma(\alpha+1) = \alpha\Gamma(\alpha)$.

We then extend the likelihood formulation to any number of focus nodes. When several focus nodes are selected, the predictive probability of a tree cut is as follows:

$$p(\mathscr{D}_{f1},\mathscr{D}_{f2},...,\mathscr{D}_{fm}|\phi) = \prod_{i \in [1,m]} p(\mathscr{D}_{fi}|\phi). \quad (9)$$

Directly maximizing the aforementioned predictive probability is intractable; thus, we adopt the tree pruning procedure presented in [20] for optimal tree cut selection.

4.3 Postprocessing

A set of representative topic nodes is selected to represent the topic tree at each time step. using the evolving tree cut algorithm. However, two issues remain. First, the resulting tree cuts may not ideally reflect user interests because a topic node can have any number of children. For example, a topic node that is highly related to the focus node can have many less-related siblings. Considering that a tree cut cannot simultaneously include a highly related node and its parent, all of its siblings have to be included in the tree cut as well. This condition results in showing less-related content with unnecessary details. Second, the number of representative topics at several time steps is still too large to be displayed in the limited screen area in many cases.

To address these issues, we propose a postprocessing approach to further reduce the topic number. This approach (1) encourages the merging of related siblings with similar content that is less related to the focus nodes, (2) discourages the merging of topics that are highly related to the focus nodes, and (3) maintains smoothness between adjacent topic sets over time.

To meet the aforementioned requirements, a clustering method is needed. Mean-shift clustering [10], which automatically determines the cluster number, can be easily adapted to fulfill all the requirements. The first requirement can be satisfied by any clustering method. Thus, we focus on how to fulfill the remaining requirements.

To meet the second requirement, an adaptive window size $\mathbf{WS}$ is defined for different clustering centers T_c.

$$\mathbf{WS}(T_c) = \begin{cases} 0 & \text{if } \mathbf{S}(\mathscr{D}_c, \mathscr{D}_f) \geq \gamma, \\ (\gamma - \mathbf{S}(\mathscr{D}_c, \mathscr{D}_f))w_{max}/\gamma & \text{otherwise.} \end{cases} \quad (10)$$

where γ is the similarity threshold, w_{max} is the maximum window size, and $\mathbf{S}(\mathscr{D}_c, \mathscr{D}_f)$ is the cosine similarity.

To meet the third requirement, all the tree cuts are generated in temporal order. Smoothness between adjacent topic sets is preserved by treating the previous clustering centers as the initial centers of the current cut node clustering.

5 VISUALIZATION

5.1 Design Rationale

We designed the *TopicStream* visualization iteratively with three domain experts, including one professor in media and communication (P1), one professor who majored in public opinion analysis in healthcare (P2), and one researcher who operates a visualization start-up (S3). These experts are not co-authors of this paper. We discussed with the experts about the analysis process and need in their work. In general, they desire a system that provides a coherent view of the evolving topics in text streams and compares incoming content with previous content. We derived the following design guidelines based on their feedback and previous research.

R1 - Providing an overview of a text stream. The experts requested a summary of old, recent, and incoming documents in the text stream. With such a summary, they can easily form a full picture of the text stream, including its major topics and their evolutionary patterns over time. In addition,

a summary was also requested to provide historical and contextual information for incoming documents. This is consistent with the design rationale of fisheye view [17]. Expert S3 commented that, "a smooth transition between new data and old data is very helpful for me to find connections."

R2 - Revealing how incoming documents merge with existing ones. Previous research into visual sedimentation [21] has shown that a smooth transition between the focus (new data) and the context (old data) helps users understand a text stream. The experts also confirmed that understanding how incoming documents merge with historical documents is useful in their analysis. For example, P1 said that, "Examining the speed, volume, and sequential order of incoming data is very useful to study agenda setting in my field."

R3 - Comparing document content at different times. Experts frequently compare the content of new documents with those of old ones in their daily analysis. For example, expert P2 commented that, "In a multi-source text stream, one source may follow another to publish documents on a specific topic. I am interested in comparing this follower-followee relationships in the new time slot with that of other time stpes, to obtain a clear understanding of who follows whom in a topic of interest." Thus the system should facilitate the visual comparison of documents at different times.

5.2 Visualization Overview

Based on the guidelines described in Sec. 5.1, we designed the *TopicStream* visualization (Fig. 5). The *x*-axis represents time. The cut nodes are visualized as vertical bars at the corresponding time step. The evolutionary relationship between cut nodes is represented by the stripes between the corresponding vertical bars. The flowing dots on the right side represent the newly arrived documents that are currently streaming in. The different colors encode various topics.

The core intent of our visualization is to help users track the dynamic characteristics of text streams. Every detail in our design was carefully crafted to cater to this purpose. For example, sedimentation animation is used to merge newly arrived documents in the dominant center of the visualization (**R2**). As the number of arriving documents increases, topic bars gradually move to the other side of the display and leave a space for new topics (**R1**). With such mechanisms, users can focus on the latest development of topics and identify interesting patterns to conduct further analysis. In particular, the visualization consists of four regions (Fig. 5, **R1**):
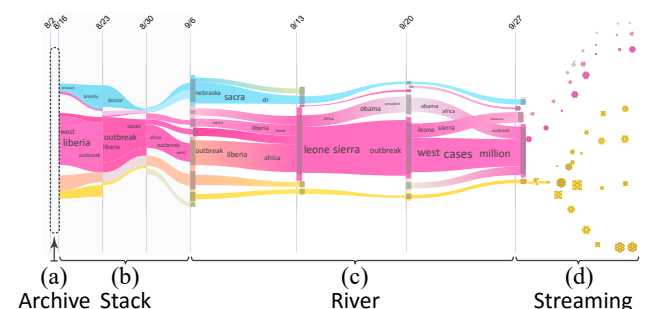

Fig. 5: The visualization is divided into four areas: (a) archive; (b) stack; (c) river; (d) streaming.

1) **Streaming**, which is on the rightmost side of the visualization, consists of newly streamed-in documents (e.g., the time period after Sep. 27 in Fig. 5(d)).
2) **River**, which is the dominant region of the visualization, consists of recent topics along with their splitting and merging relationships (e.g., Sep. 6 - 27 in Fig. 5(c)).
3) **Stack**, which is to the left of the river region, contains older topics and documents (e.g., Aug. 16 - Sep. 6 in Fig. 5(b)). To reduce the visual complexity caused by the splitting and merging relationships, this region removes splitting/merging branches and only displays the mainstream of each topic. Since users want to keep track of how the topics in this region connected with the topics in the river region, the white spaces between the topic stripes are not removed. The width of each time step in this region is smaller than that in the river region to save space.
4) **Archive**, which is on the leftmost side, contains the oldest topics and documents (e.g., Aug. 2 - 16 in Fig. 5(a)). Although the stacked region can reduce the amount of space required, it is still cluttered for a text stream with tens or even hundreds of time steps. To solve this issue, we introduce the archive region, which uses a stacked bar (Fig. 5(a)) to represent documents whose times are k time steps earlier than the newly streamed-in ones. In *TopicStream*, k is specified by the user. For example, k is set to 8 in the example of Fig. 5. To save space, the width of the bar is fixed no matter how many documents are archived. Each bar item represents a topic. Its height represents the average number of documents of each time step that belongs to this region.

As described above, the visualization designs for a bar and a stacked graph are quite straightforward. We will next introduce the visualization designs of the river and streaming regions in detail.

5.3 Visualization Design

5.3.1 Tree Cut as a River

Visual Encoding. Each cut node is represented by a vertical bar (topic bar) similar to that presented in [12]. The tree depth of a cut node is represented by the horizontal offset to the time step. When a node in the tree is deep, the corresponding topic bar moves to the right.

The number of documents contained in a topic node is represented by the height of the topic bar. The width of the colored stripe between two topic bars indicates the number of document pairs between the two bars. For example, the left width of the stripe represents the portion of documents mapped to the documents in the right topic bar. The dark region in the middle of a topic bar represents the portion of documents mapped to the documents both in the previous and the next topic trees (Fig. 5).

Layout. The basic representation of the visualization is a directed acyclic graph (DAG). A node represents a topic and an edge between nodes encodes the evolutionary relationships between topics with mapping. When a new batch of documents is processed, we first run the DAG
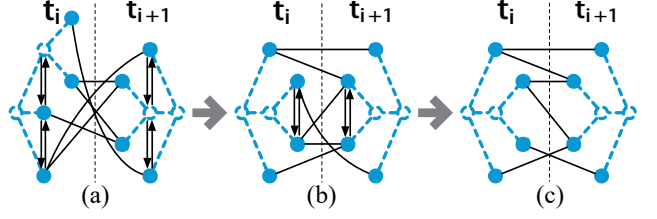

Fig. 6: Reordering example: (a) reorder level one; (b) reorder level two; (c) result.
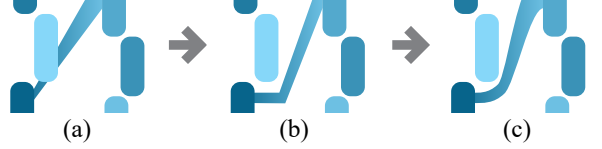

Fig. 7: Example of edge routing: (a) the stripe is hidden by the topic bar; (b) two intermediate points are added; (c) a Bézier curve is utilized to improve visual quality.

layout algorithm to determine an optimal order for the new topic nodes. Once the topological structure is computed, a force model is built to generate the sedimentation animation and merge new documents with existing topic bars.

We initially reorder the cut nodes at each time step to minimize edge crossings between neighboring time steps and generate a legible layout that illustrates the evolving patterns. Edges are then routed to avoid overlapping between nodes and edges. Finally, representative documents are packed on a selected stripe.

Reordering. Sugiyama's heuristics [31], which is a well-known DAG layout algorithm, is employed to reorder the nodes at each time step to minimize edge crossings. However, if we directly run the algorithm without constraints, sibling nodes can be separated by other nodes. We implement Sugiyama's heuristics from the highest to the lowest levels of the tree at each time to ensure that the sibling nodes stay together. Fig. 6 provides an example generated by the reordering algorithm.

Edge Routing. Stripes and topic bars can overlap because topic nodes are offset to encode their depth (Fig. 7(a)). We employ the edge routing technique [13] to solve this problem. Two additional intermediate points are introduced for each overlapping part to route the stripe. The Bézier curve is utilized to help users follow the striped path (Fig. 7).

Packing. We pack the documents on the topic stripe (**R3**) to help users understand and compare their relationships, including the incoming order and similarity relationships. Each news article is represented by a circle in our visualization, whereas each tweet is represented by a square. For the sake of simplicity, each square is approximately represented by a circle whose center is the same as the square's and whose radius is $\beta \cdot \sqrt{2}b$. b is the side length of the square and β ($1/\sqrt{2} \leq \beta \leq 1$) is a parameter that balances the intersection and gaps between elements (e.g., circles and squares) in the final packing result. The larger β is, the larger the gap might be. The packing problem is formulated as a circle packing problem using this approximation. We then employ a front-chain-based circle packing algorithm,
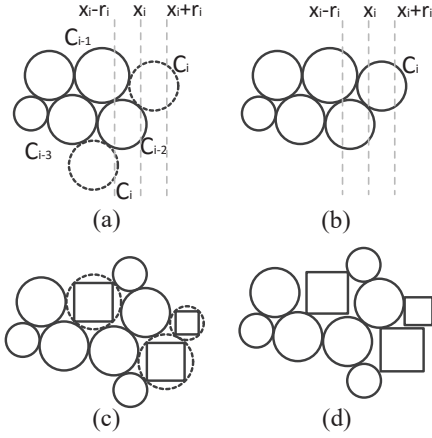
Fig. 8: Illustration of the packing algorithm: (a) finding possible placement positions of C_i; (b) setting the position closest to $(x_i, 0)$ as the placement position; (c) replacing several circles with the corresponding squares; (d) reducing the gap with the size constraints and deriving the final packing result.

as in [35], [46], to pack circles tightly on the selected stripe. Fig. 8 illustrates the basic idea of this packing algorithm.

Compared with the packing problem described in [46], our problem does not provide the initial x coordinate for each circle. Only the incoming order of each circle is provided in our packing problem. Thus, we have to derive the initial x coordinate based on the order of the circles. The basic idea is to determine an approximate placement position for each circle, which is achieved by approximately mapping its area to the area of the segmented stripes. The average of the x coordinates of the corresponding segmented stripes is then used to approximate the initial x coordinate of the circle. In particular, we align all the circles on a straight line based on their areas (Fig. 9(a)). The area of circle C_i is πr_i^2. We then divide the stripe into n uniform segments along its x-axis. The height of the k-th segment is denoted as h_k and its area is wh_k, where w is the width of each segment along the x-axis. All these segments are also aligned on a straight line based on their areas (Fig. 9(b)). Fig. 9 shows that the overlapping relationship between the area of the circle and that of the segmented stripes can be determined using two straight lines. For example, the initial x_i of circle i in this figure is approximated by $average(u_k, u_{k+1}, u_{k+2})$. Here u_k is the x coordinate of the center of k-th segment.

Interaction. We also provide the following interactions to
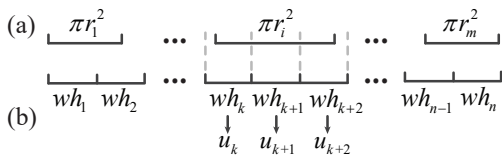

Fig. 9: Deriving the initial x position: (a) align all the circles on a straight line based on their areas; (b) align all the stripe segments on a straight line based on their areas. The dotted vertical lines indicate the overlapping relationship between the area of the circle and that of the segmented stripes. Based on this relationship, x_i is approximated as an $average(u_k, u_{k+1}, u_{k+2})$.

explore the complex evolutionary clustering results from multiple perspectives in addition to the interactions described in [12] (e.g. details on demand, collapsing/expanding time steps, splitting/merging topic bars, and changing focus).

Document Query. Once the documents are transformed into a colored stripe, we adopt circle packing to encode the documents contained within the color stripe for further query and analysis. The example in Fig. 10 shows how users can click the stripe and turn it into a circle/square packing, in which a circle represents a news article and a square encodes a tweet. Once the packing result is displayed, users can manually click one or more documents to examine the content in detail.

Visual Comparison. We allow users to compare the relationships among different time steps by leveraging a circle packing algorithm. For example, users can compare the incoming order and similarity relationships, as shown in Fig. 19(a). One of our experts, P2, commented that, "Comparing the incoming order of documents helps me easily discover who talked about a topic first (that is, who set the agenda) and who immediately followed. This feature can help me study agenda setting in my field."

5.3.2 Streaming Document as Sedimentation

Visual Encoding. Inspired by visual sedimentation [21], we use the river sedimentation metaphor to encode the process of newly arrived text documents that merge with existing topics (**R2**). To quicken the sedimentation process of a high-volume text stream, a set of document clusters are derived from the incoming documents by using k-means clustering. A token is a visual mark representing a document cluster. The generation process of the sedimentation metaphor consists of four steps:

Entrance. Newly arrived documents are represented as circular or rectangular tokens (Fig. 5) that come into view from the right side. Documents with similar content are clustered into one token, the size of which indicates the number of documents, to handle the scalability issue. The color of each token encodes the topic that it contains.

Suspension. Each token moves toward (from right to left) the corresponding topic bars of the latest time step. Token size decreases gradually during the movement.

Accumulation and *decay*. The tokens will stop moving and start to decay once they touch the corresponding topic bars or other tokens that have already settled. The settled tokens continue to shrink and merge with existing topics.

Aggradation. The colored stripes continue to grow and indicate the latest development of topics when the settled tokens are resolved.
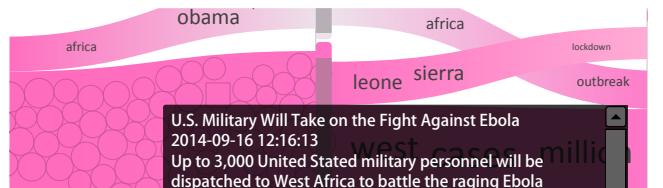

Fig. 10: Encode documents after sedimentation as circle/square packing.

Fig. 11: Relevant documents of cluster A are highlighted in the river (B), the stack (C), and the archive (D) regions.

Once a batch of documents (e.g., for a day) are sedimented, the corresponding topic bars appear and push older topic bars to the left-hand side. The archive and stack regions then change accordingly.

Layout. Each token is assigned to a region based on the topological structure in the "reordering and edge routing" step during the sedimentation process. The token can only move within the assigned region and cannot cross the border. The speed of the token is controlled by two forces: 1) a universal gravity force and 2) an attractive force between the token and the sedimented tokens. The gravity force provides each token with constant acceleration from right to left. The attractive force ensures that similar documents will sediment close to one another. Therefore, the total acceleration a_k for a moving token k is defined as

$$a_k = g + \sum_i s_{ik} * n_i / ||p_i - p_k||^2,$$

where g is the constant gravity acceleration, p_i is the location of sedimented token i, p_k is the location of token k, n_i is the number of documents in token i, and s_{ik} is the content similarity between tokens i and k.

Interaction. The sedimentation visualization also allows users to examine the content of the incoming documents interactively and compare them with older documents.

Document Link. In many text stream analysis tasks, it is desirable to quickly find related documents covering a long time period. Document link is supported for this requirement of our system. For example, users can initially explore the content in the streaming region and find a document/cluster of interest. Our system then automatically uses the word vector in the given document and locates the most similar documents in all three regions (i.e., streaming, stack, and archive). Once the related documents/clusters are located, the connections are displayed for users to explore further.

An example of document link is shown in Fig. 11, in which a user explores relevant documents from an incoming Twitter cluster (Fig. 11A). Relevant documents are found in the river (Fig. 11B), stack (Fig. 11C), and archive (Fig. 11D) regions. The archive region is expanded accordingly to facilitate the examination of the relevant documents.

Users can also click on a token while it is still in the suspension step. Related documents are then displayed for further examination.

6 QUANTITATIVE EVALUATION

In this section, a quantitative evaluation of the proposed streaming tree cut algorithm is conducted.

6.1 Fitness and Smoothness

To assess the effectiveness of the streaming tree cut algorithm, we compared our algorithm with a baseline algorithm in terms of fitness and smoothness.

6.1.1 Criteria

Fitness and smoothness are two important criteria to evaluate the derived streaming tree cuts. Fitness measures how satisfactorily the topics on the tree cut represent the topic distribution within a topic tree.

Fitness (F): We derived the measure from the proposed tree cut likelihood equation, $F = p(\phi^t|T^t)p(\mathscr{D}_f|\phi^t)$, where the right side is defined in Eq. (2). $p(\phi^t|T^t)$ describes how the tree cut fits the tree and $p(\mathscr{D}_f|\phi^t)$ describes how it fits the focus. A larger F value indicates a better tree cut.

The following three measures assess the smoothness between the adjacent tree cuts. In the implementation, a larger smoothness value means that the two adjacent tree cuts are smoother.

Tree mapping (S_{map}): The measure is derived from the smoothness cost function of the streaming tree cut algorithm, $S_{map}(\phi^t, \phi^{t-1}) = -E_2(\phi^t, \phi^{t-1})$, where $E_2(\phi^t, \phi^{t-1})$ is defined in Eq. (4).

Normalized Mutual Information (NMI) (S_{NMI}): The NMI measure represents the mutual information shared by both the cluster assignments and a pre-existing label. The Hungarian algorithm [30] is employed to find the optimal match between the document sets of the two tree cuts. This measure assesses the similarity between adjacent tree cuts.

Tree distance (S_{dist}): This measure is used to evaluate the difference between the tree cuts by aggregating the tree distance between two related cut nodes T_s and T_r,

$$S_{dist}(\phi^t, \phi^k) = - \Big(Avg_{T_r, T_s \in \mathscr{C}_{\phi^t}} (D_{T^t}(T_r, T_s) - D_{T^k}(T_r, T_s))^2$$
$$+ Avg_{T_r, T_s \in \mathscr{C}_{\phi^k}} (D_{T^k}(T_r, T_s) - D_{T^t}(T_r, T_s))^2 \Big) /2, \quad (11)$$

where $D_T(T_r, T_s)$ is the tree distance between T_r and T_s under T. If T_r and T_s are not in T, they are mapped to T.

6.1.2 Experimental Settings

A baseline system was implemented according to the DOI-based tree cut generation method [12]. To compare the fitness and smoothness of the proposed methods to the baseline, we conducted experiments on the following two datasets.

- **Dataset A** contains 207,406 news articles and 15,565,532 tweets related to "Ebola" (from Jul. 27, 2014 to Feb. 21, 2015). The articles were organized into 30 topic trees by week. The tree depth, total node number, and first-level node number of the trees varied from 3 to 5, 34 to 223, and 10 to 33, respectively.
- **Dataset B** contains 543,114 news articles related to "Obama" (from Oct. 14, 2012 to Feb. 21, 2015). The articles were organized into 62 topic trees by every two weeks. The tree depths varied from 4 to 11, the total node numbers changed from 246 to 471, and the node number of the first level ranged from 18 to 79.

TABLE 2: Evaluation of the overall likelihood and smoothness.

$f_r(\cdot) = \frac{m_o - m_b}{m_b} * 100\%$, where m_b and m_o are the measure values of the baseline method and our method.

Dataset	$f_r(\boldsymbol{F})(\%)$	$f_r(\boldsymbol{S}_{map})(\%)$	$f_r(\boldsymbol{S}_{NMI})(\%)$			$f_r(\boldsymbol{S}_{dist})(\%)$		
	$F(\phi^t)$	$S_{map}(\phi^t,\phi^{t-1})$	$S_{NMI}(\phi^t,\phi^{t-1})$	$S_{NMI}(\phi^t,\phi^{t-2})$	$S_{NMI}(\phi^t,\phi^{t-3})$	$S_{dist}(\phi^t,\phi^{t-1})$	$S_{dist}(\phi^t,\phi^{t-2})$	$S_{dist}(\phi^t,\phi^{t-3})$
A	4.5486	18.7873	7.6839	-1.8875	-2.6680	13.5781	-1.5226	-2.4843
B	7.5084	26.4451	8.7131	-3.1711	-3.2452	13.1334	-4.2689	-5.4008
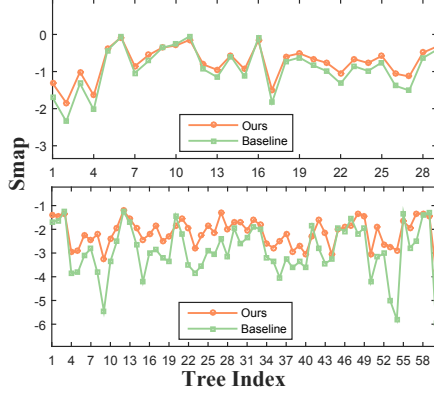
Fig. 12: Comparison of tree mapping smoothness.

To eliminate bias caused by the focus node selection, the same number of focus nodes was randomly selected 50 times and the experiments were repeated 50 times. At each time, F for each tree cut was computed. Since the measure S_{map} was defined on adjacent tree cuts, we only computed S_{map} between adjacent tree cuts. To demonstrate the global smoothness of the proposed algorithm, S_{NMI} and S_{dist} were

computed between ϕ^t and each of ϕ^{t-1}, ϕ^{t-2}, and ϕ^{t-3}. The results were computed by averaging the 50 trials.

6.1.3 Results

The overall fitness and smoothness were compared with the baseline. As shown in Table 2, the proposed method generates a much smoother structure than the baseline while maintaining greater fitness. When the smoothness between non-adjacent tree cuts was compared, the proposed method performed slightly worse, because the method only considered the adjacent tree cuts to improve the performance of the data stream. Thus, the global smoothness was not maintained to a certain extent.

We further compared the smoothness of our method with the baseline between trees under these measures. As shown in Figs. 12, 13, and 14, the proposed streaming algorithm worked as well as the baseline under the three measures for adjacent tree cuts. For non-adjacent tree cuts, the smoothness of the proposed algorithm was slightly worse under the commonly used measures NMI and tree distance. The fitness of the proposed algorithm at each tree was also evaluated.
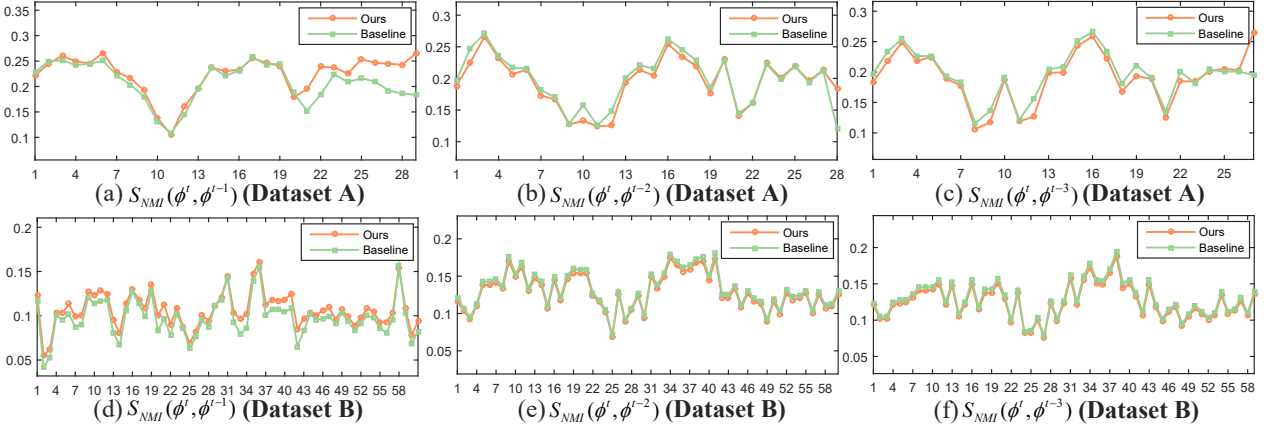
Fig. 13: Comparison of NMI smoothness. X-axis represents tree index and Y-axis encodes NMI smoothness.
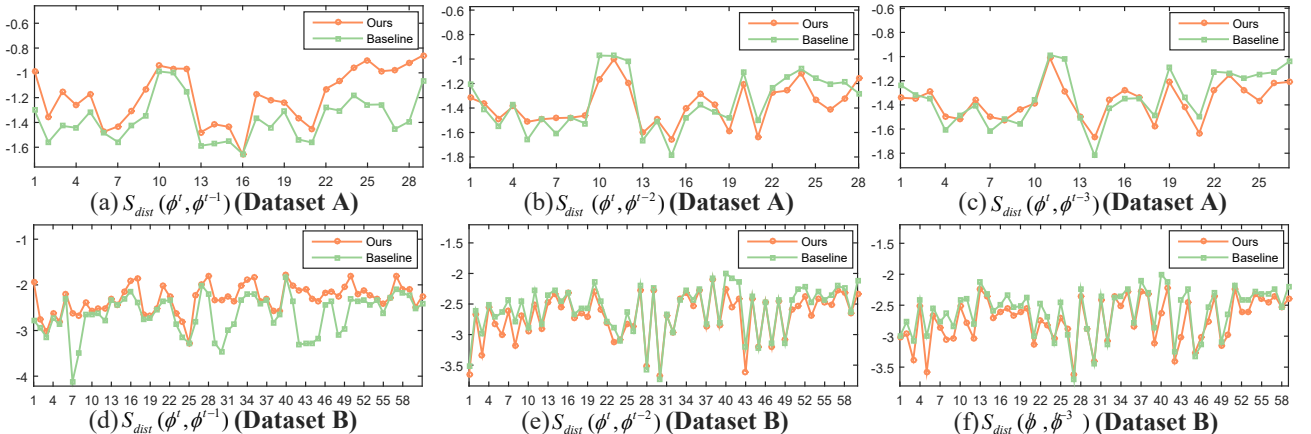

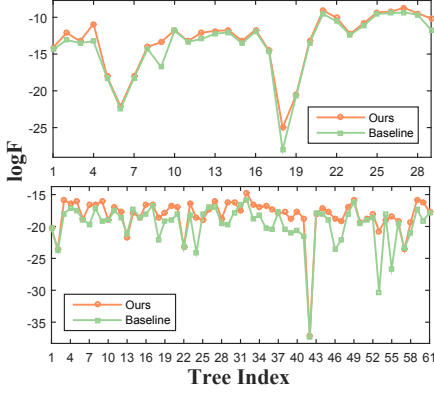Fig. 14: Comparison of tree distance smoothness. Y-axis encodes tree distance smoothness.

Fig. 15: Comparison of fitness at each tree.

As shown in Fig. 15, the proposed algorithm was more effective than the baseline at each time in all the datasets. These findings demonstrate that the proposed algorithm can preserve the smoothness between the adjacent trees as well as the fitness without sacrificing global smoothness.

6.2 Scalability

We conducted two experiments to evaluate the scalability of our algorithm. In the first experiment, we investigated the ability of our algorithm to handle topic trees with a large number of internal nodes (I_{num}). In the second, we tested the ability of our algorithm to process long sequences of topic trees.

6.2.1 Experimental Settings

The dataset used in the first experiment was generated by copying the first ten trees in Dataset A s times ($s \in \{1, 3, ..., 15\}$). As a result, we obtained eight groups of topic trees with varied I_{num} ($I_{num} \in \{118, 354, ..., 1770\}$). For each group of topic trees, we treated the first five trees as old trees and evaluated the average time to process the 6th to 10th trees. In our experiments, focus nodes were randomly selected to avoid any biased conditions. To eliminate randomness caused by the focus node selection, we randomly selected the given number m ($m \in \{1, 3, 5\}$) of focus nodes 50 times and ran the experiment 50 times. Results were computed by averaging the 50 trials.

In the second experiment, we used the 30 topic trees in Dataset A. Specifically, we regarded the first P_{num} ($P_{num} \in \{7, 9, ...29\}$) trees as old trees, and evaluated the time to process the ($P_{num} + 1$)-th tree. All other settings were the same as the first experiment.

The experiments were run on a workstation with an Intel Xeon E5-2630 CPU (2.4 GHz) and 64GB Memory.

6.2.2 Results

As shown in Fig. 16, the running time of our algorithm increases at an approximate quadratic rate with the increase of I_{num}. For $m = 5$, our algorithm can process topic trees with 1,770 internal nodes in 66 seconds. This demonstrates that our algorithm can handle large topic trees.

Next, we demonstrated the scalability of our algorithm in regards to P_{num} under different m. We used a normalized
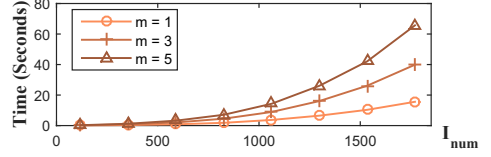

Fig. 16: Running time vs. number of internal nodes in the topic tree (I_{num}) vs. number of focus nodes (m).
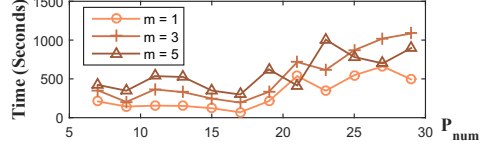

Fig. 17: Normalized running time vs. number of topic trees processed (P_{num}) vs. number of focus nodes (m).

running time to eliminate any bias caused by different sizes of the topic trees. Normalized running time is calculated by multiplying real running time with (I_{avg}/I_{cur})^2. Here I_{avg} is computed by averaging I_{num} of all trees and I_{cur} is I_{num} of the ($P_{num} + 1$)-th topic tree. As shown in Fig. 17, the normalized running time increases slowly with the increase of P_{num} and the results are consistent across different m. This indicates that our algorithm can process long sequences of topic trees efficiently.

7 CASE STUDY

In this section, we demonstrate the usage scenarios of our approach using real-world datasets.

7.1 Ebola Data

The case study was conducted with a professor (P2) who majored in public opinion analysis in healthcare. In this case study, we illustrate how TopicStream helps an expert examine the relationship between the severity of an epidemic (e.g., Ebola) and the intensity of public opinion. The severity of the epidemic was measured by the reported number of cases and deaths. The intensity of public opinion was represented by the number of news articles and tweets at that time step (the width of the topic stripe). A wider stripe indicated more intense public opinion (Fig. 18).

A dataset that contains both news articles and tweets collected by using the keyword "Ebola" was used (Dataset A). Table 3 shows the statistics of the dataset.

Data	Time span	N_{num}	T_{num}	h	I_{num}
Old	7/27/2014-9/27/2014	51,318	7,161	3-4	77-150
New	9/28/2014-2/21/2015	156,088	15,558,371	3-5	34-223

TABLE 3: Statistics of the Ebola dataset. Here N_{num} denotes the number of news articles, T_{num} represents the number of tweets, h is the tree depth, and I_{num} denotes the number of internal nodes in the tree.

Spread of Ebola outbreak. We first provided the professor (P2) with an overview of the old Ebola data. The old data (before Sep. 27, 2014) is shown in Fig. 18(a). The news articles from Sep. 28 to Oct. 4 appeared in a streaming manner, as shown in Fig. 18(b). Using topic keywords and corresponding news articles in Fig. 18(a), P2 immediately
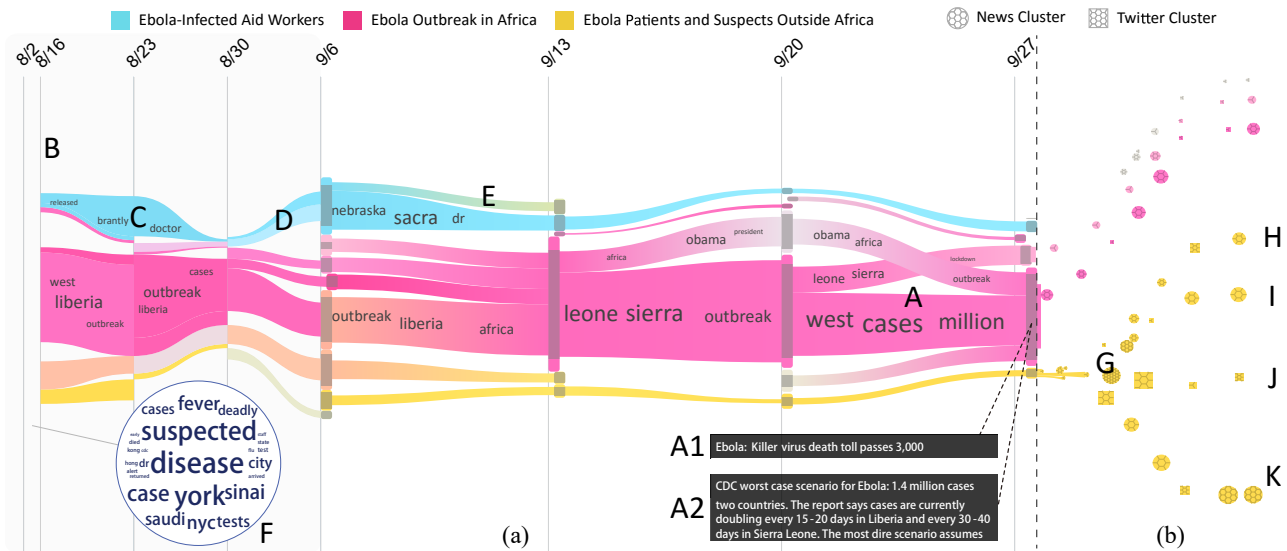
Fig. 18: Comparative analysis between the severity of the epidemic and the intensity of public opinion in the Ebola dataset. (a) The region with the most severe cases (i.e., Africa) was the key focus of public opinion before Sep. 27, 2014. Epidemic severity during this period was consistent with public opinion intensity. (b) An explosive growth of public discussion in non-African regions occurred after Sep. 27. Public opinion intensity during this period was inconsistent with epidemic severity.

identified the major topics in the news stream, which were encoded as blue, pink, and yellow. As in [12], we used the mean-shift clustering algorithm to cluster the topic at the first level since it is the most abstract level and can represent the topic tree very well. For each cluster, we chose the topic closest to the cluster center as one focus topic.

By examining the incoming news articles on the pink topic stripe, "Ebola outbreak in Africa" (Fig. 18A), P2 found that the epidemic was extremely serious in Africa. The epidemic caused a large number of deaths (Fig. 18A1) and the spread of infections was rapid. For example, the news article entitled "CDC worst case scenario for Ebola: 1.4 million cases" mentioned that reported cases in Liberia were doubling every 15 to 20 days and those in Sierra Leone were doubling every 30 to 40 days (Fig. 18A2). The blue topic stripe contains keywords "dr," "sacra," and talks about "Ebola-infected aid workers." "Sacra" is the last name of Dr. Rick Sacra, one of the aid workers. By examining the news articles in the archive area (Fig. 18B), P2 learned that two aid workers returned to the U.S. for treatment. The increased width in the stack area (Fig. 18C) discussed their recovery. Figs. 18D and 18E in the river area are related to the third and fourth infected aid workers. From the preceding exploration, P2 concluded that several aid workers had been infected; however, the situation was not serious. From keywords "suspected," "york," and "sinai" in the word cloud of the yellow topic stripe (Fig. 18F)), P2 concluded that this topic was about "Ebola patients and suspects outside Africa." After reading the corresponding news articles before Sep. 27, P2 concluded that only a few suspects were outside Africa and the situation was not serious.

Explosive discussion on Ebola outside Africa. P2 found that the severity of the epidemic was consistent with the intensity of public opinion before Sep. 27 (Fig. 18(a)), that

is, the stripe was wider, which indicated more intense public opinion. However, as indicated by the increasing number of yellow circles and squares in the visualization (Fig. 18(b)), there is an explosive discussion on Ebola outside Africa occurred after Sep. 27. P2 was curious about such a change; thus, the exploration of the incoming data continued.

She noticed that the explosion began at the news cluster denoted by Fig. 18G, which contained many news articles. The news cluster was then followed by several Twitter clusters. After some exploration, P2 found that the news cluster was mainly about the first case of Ebola in the US. The patient, Thomas Duncan, had been exposed to as many as 80 people. The first confirmed case led to numerous discussions on Twitter and created fear. Because of the heightened attention from the public and media, this topic was divided into four sub-topics: 1) the further report of suspects (Fig. 18H); 2) government actions (Fig. 18I); 3) treatment of the patient (Fig. 18J); and 4) the search for people who had some form of contact with the patient (Fig. 18K).

P2 commented that the public in the US paid minimal attention to the Ebola epidemic in Africa before Sep. 27, observing the epidemic from the other's perspective. This is consistent with the theory of alterity (otherness) [1]. When the epidemic arrived in the US, the perspective changed and led to intense discussions on news media and Twitter. P2 further explained that the spread of the first case in the US also disclosed another phenomenon. Since the news media reported the first case wantonly, the severity of the epidemic was overestimated and fear was created among average people. This is because human perception is often influenced by the pseudo society built by the media. Under such a situation, the government must guide public opinion. **Action and guidance of the government.** P2 continued examining new documents to learn the actions of the gov-
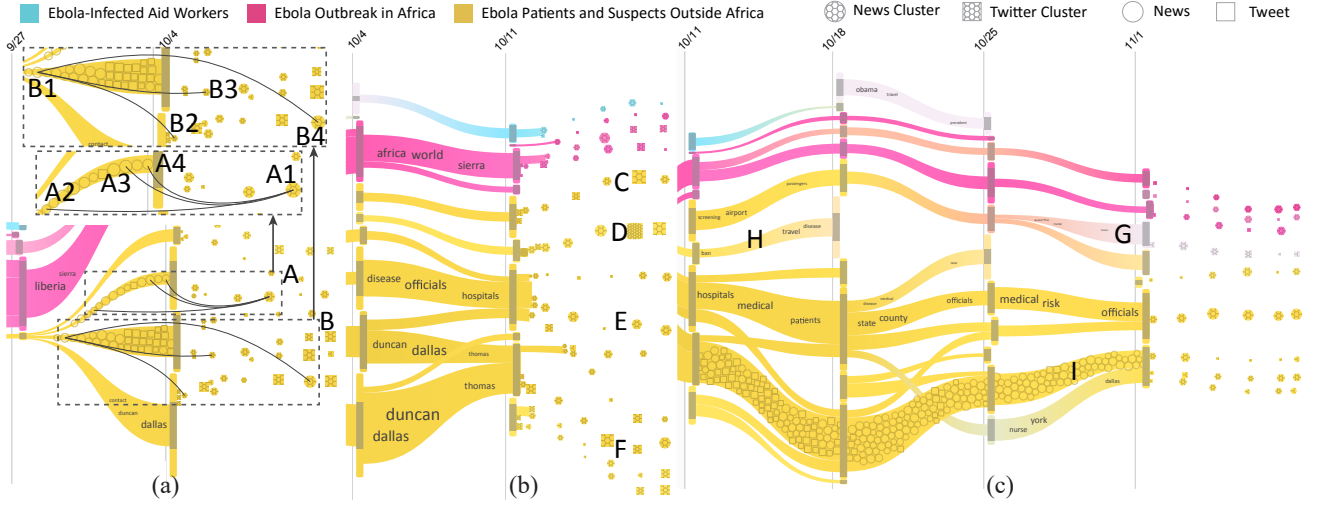
Fig. 19: Explosive discussion of reported cases outside Africa: (a) Oct. 5 to 11; (b) Oct. 12 to 18; (c) Nov. 2 to 8.

ernment regarding the epidemic. She found few discussions on Twitter on the topic "government actions" from Oct. 5 to Oct. 11 (Fig. 19A), indicating that this topic failed to attract public attention. On the contrary, numerous discussions on Twitter focused on the death of an Ebola patient (Fig. 19B).

To identify the reason, P2 examined these two topics in Fig. 19(a) and found one representative cluster (Fig. 19A1, "EbolaCDC urges hospitals to follow Ebola-related protocols") and document (Fig. 19B1, "Dallas hospital isolating patient being tested for Ebola"). She then explored similar documents or clusters in the adjacent time steps to determine the evolution of the topics in the stream. From the links and corresponding documents in Fig. 19A, P2 found that the government immediately took action and prepared for Ebola before Oct. 4, as shown by the following news articles: 1) Sep. 30, "Health Ministry to distribute 10,000 PPEs on Thursday" (Fig. 19A2); 2) Oct. 2, "Local hospitals prepared in case of Ebola" (Fig. 19A3); and 3) Oct. 2, "Ebola 'unlikely' but South prepared" (Fig. 19A4). From the links and corresponding clusters in Fig. 19B, P2 realized that the patient's condition worsened and led to death, as indicated by the following news articles: 1) Oct. 5, "Dallas Ebola patient is in critical condition, hospital says" (Fig. 19B2); 2) Oct. 5, "Ebola patient in Dallas takes a turn for worse" (Fig. 19B3); and 3) Oct. 8, "Dallas Ebola Patient Dies" (Fig. 19B4). P2 believed that the government acted promptly. However, the death of the patient led to uncontrolled public opinion.

The documents that subsequently streamed in were from the week of Oct. 11. As shown in Fig. 19(b), public attention to the topic "government action" decreased (Fig. 19E), whereas discussions on Twitter on topics "airport screening" (Fig. 19C, "Ebola Screenings Begin at US Airports"), "travel ban" (Fig. 19D, "RT @CronkiteSays: VIEWER POLL#N#Do you support a travel ban from Ebola inflicted countries?"), and "infected nurse" (Fig. 19F, "Dallas Nurse With Ebola Identified") increased. The public paid more attention to negative messages. The situation improved after three weeks (Fig. 19(c)). P2 analyzed the documents and specified three reasons for this change. First, the change from topic "airport screening" to another topic shifted

public attention (Fig. 19G). The new topic was related to quarantine, which emerged because nurse Kaci Hickox defied the quarantine imposed on her after returning from treating Ebola patients in West Africa. This event caused great disturbance and shifted public attention. Second, topic "travel ban" gradually disappeared after President Obama decided to cancel the travel ban (Fig. 19H). Third, the popularity of "infected nurse" gradually decreased as the nurse was cured and returned to normal life (Fig. 19I). By now, fear caused by the first case of Ebola in the US disappeared and the government finally influenced public opinion to be more positive. P2 indicated that the government was successful in using another topic ("quarantine", Fig. 19G) to shift public attention away from the negative opinion caused by the first Ebola case in the US.

7.2 Obama data

The second case study was a collaboration with a professor (P1) of media and communications. In this case study, P1 studied the relationship between the media agenda (mass media) and public opinion, a long-standing research topic in the field of media and communication [29].

We used a news dataset collected by using the keyword "Obama" (Dataset B), which is summarized in Table 4.

To analyze the relationship between the media agenda and public opinion, several pieces of contextual information were added (the dashed rectangle in Fig. 20(a)). The contextual information consists of: 1) Obama's presidential approval rating, 2) an economic confidence index derived from Gallup public opinion polls [3], and 3) a time-varying statistical correlation between the Gallup poll results and the sentiment of media articles.

A word-embedding-based sentiment classification algorithm [34] was employed to calculate the sentiment for each

Data	Time span	N_{num}	h	I_{num}
Old	10/14/2012-12/8/2012	47,963	4-10	267-376
New	12/9/2012-2/21/2015	495,151	7-11	246-471

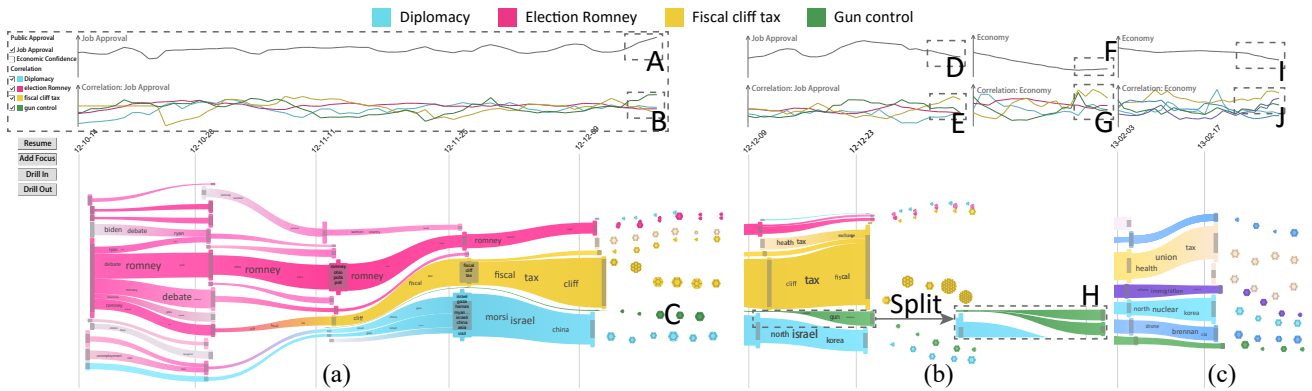TABLE 4: Statistics of Obama dataset.

Fig. 20: Significant changes in public opinion in the Obama dataset. (a) Presidential approval rating affected by topic "Gun Control." (b) A decrease in presidential approval and economic confidence caused by the fiscal cliff crisis. (c) Another low economic confidence rating caused by failed negotiations on government spending cuts.

article. The topic "sentiment" at each time step refers to the average sentiments of the documents at that time step. A sentiment time series was then obtained for each topic. Finally we calculated the Pearson correlation coefficient between a Gallup poll result and the temporal sentiment of a topic. **The presidential approval rating affected by the topic "gun control."** In the old data (Fig. 20(a)), P1 detected four different topics: "diplomacy" (blue), "election" (red), "fiscal cliff and taxes" (yellow), and "gun control" (green). He then started the analysis from Dec. 9, 2012, which was just before the formal re-election of President Obama. P1 observed an increase in the curve for the presidential approval rating (Fig. 20A). By comparing the correlation between this index and the sentiment curve of each topic, he found that the highest correlation was with the topic "gun control" (Fig. 20B). This topic received much more attention than before the week of Dec. 9 (Fig. 20C), which was triggered by a shooting massacre at a Connecticut elementary school on Dec. 14, 2012. To examine how people responded to this incident, P1 split this topic and found two subtopics (Fig. 20H). One is the president's response and the other is the response of others (congressional representatives, NRA, and the public). P1 found that the public called for tighter gun control ("Gun-control petition to White House breaks record"). Obama's actions fit with public opinion very well ("Obama vows to battle gun violence"). We speculate that this was the major cause for the increase in his approval rating.

Public attention transition to topic "fiscal cliff and taxes." P1 observed an immediate decrease in the presidential approval rating on Dec. 31, 2012 (Fig. 20D). The correlation between presidential approval and the topic "gun control" decreased to a smaller value (0.12), whereas its correlation with topic "fiscal cliff and taxes" increased to its highest (0.51, Fig. 20E). P1 explained that this topic was about the fiscal cliff crisis at the end of 2012. The government faced an act that would take effect on Jan. 1, 2013. Large tax increases and spending cuts were included in this act. To postpone this act, the president and the two political parties debated for a long time and settled on a temporary solution on Jan. 1. They agreed to postpone the spending cuts until Mar. 1. After reading the news, P1 found that people surmised that the president did not truly want

the crisis to end. As this topic concerned the economy, P1 considered the economic confidence index. Unsurprisingly, a local minimum on Dec. 31, 2012 (Fig. 20F) was found. The low confidence level was possibly caused by raising tax rates because the correlation between this topic and the economic confidence was at its highest (Fig. 20G).

As the spending cuts were postponed to Mar. 1, P1 decided to continue tracking this event. He learned that this act would have a significant effect on the economy ("Bernanke: sequester cuts slow economic recovery"). The spending cuts took effect on Mar. 1. On this date, P1 observed another local minimum in the economic confidence (Fig. 20I). The correlation between the economic confidence and the topic "fiscal cliff and taxes" was at its highest (Fig. 20J), which was in accordance with P1's expectation. He commented that the streaming visualization was visually appealing and practically useful for examining real-time documents.

Carry-over effect of topic "fiscal cliff and taxes." P1 wanted to follow the subsequent development of this topic. He found that the topic "fiscal cliff and taxes" (yellow) appeared again on Dec. 7, 2014 (Fig. 21). This topic concerned tax breaks at the end of 2014. At this time, the economic confidence index experienced a remarkable increase (Fig. 21A). Because the correlation between this index and the topic "fiscal cliff and taxes" was at its highest (0.44, Fig. 21B), P1 speculated that intense discussions on tax breaks concerning this topic were a potential reason. P1 was curious about the significant influence of this small topic on economic confidence. To this end, P1 linked the largest document cluster (Fig. 21C) at this time to the previous relevant documents. Several documents appeared during the period of the fiscal cliff crisis in 2012 (Fig. 21D). At that time, the government wanted to raise taxes because of the fiscal cliff and this topic was dominant in the media (yellow topic in Figs. 20(a) and (b)). P1 commented that this fact could be regarded as a carry-over effect [2] in the field of media and communication. P1 further explained, "The fiscal cliff crisis left a profound impression on the public and had a great influence on the economic confidence at that time. As a result, this influence can be carried over to the relevant topic later even if it is a smaller one."

Fig. 21: Carry-over effect of media agenda: the documents on the tax increase in 2012 are connected to tax breaks in 2014.

8 DISCUSSION AND FUTURE WORK

In this paper, we have presented a novel visual analytics system to help users explore and understand hierarchical topic evolution in high-volume text streams. Powered by the streaming tree cut model and the corresponding visualization, the system allows users to analyze hierarchical topics at different granularities, as well as their evolution patterns over time. In addition, it enables users to interactively customize and refine the visualization based on their interests. A quantitative evaluation and two case studies were conducted to demonstrate the effectiveness and usefulness of the system for text stream analysis.

Although the system performs well when analyzing the evolution of hierarchical topics, it can still be improved. First, one component of our system, the evolutionary tree clustering algorithm, is effective in constructing a sequence of topic trees with high fitness and smoothness. However, relying solely on the optimization results is not always effective because the tree clustering algorithm may be imperfect and different users may have different requirements. Studying how to leverage the domain knowledge of a user in our system and allow him/her to express and define information requirements can help solve the aforementioned problem. This noteworthy topic can be pursued in the future. Second, we only utilized the horizontal offset to encode tree depth but ignored the general structure of a tree. However, users may want to examine each tree structure and obtain a complete overview of evolving topic trees in several cases. We also plan to enable tree exploration in the next version of the system and allow users to explicitly explore topic hierarchical structures.

9 ACKNOWLEDGEMENTS

REFERENCES

[1] Alterity. http://en.wikipedia.org/wiki/Alterity.
[2] Carry-over effect. http://en.wikipedia.org/wiki/Experimental_psychology.
[3] Gallup. http://www.gallup.com.
[4] A. Ahmed and E. P. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SDM*, pages 219–230, 2008.
[5] A. Ahmed and E. P. Xing. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. In *UAI*, pages 20–29, 2010.
[6] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.
[7] C. Blundell, Y. W. Teh, and K. A. Heller. Bayesian rose trees. In *UAI*, pages 65–72, 2010.
[8] C. Blundell, Y. W. Teh, and K. A. Heller. Discovering non-binary hierarchical structures with Bayesian rose trees. In *Mixture: Estimation and Applications*. John Wiley & Sons, 2011.
[9] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD*, pages 554–560, 2006.
[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE PAMI*, 24(5):603–619, 2002.
[11] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *IEEE TVCG*, 17(12):2412–2421, 2011.
[12] W. Cui, S. Liu, Z. Wu, and H. Wei. How hierarchical topics evolve in large text corpora. *IEEE TVCG*, 20(12), 2014.
[13] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE TVCG*, 14(6):1277–1284, 2008.
[14] W. Dou, X. Wang, R. Chang, and W. Ribarsky. Paralleltopics: A probabilistic approach to exploring document collections. In *IEEE VAST*, pages 231–240, 2011.
[15] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *VAST*, pages 93–102, 2012.
[16] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE TVCG*, 19(12):2002–2011, 2013.
[17] G. W. Furnas. Generalized fisheye views. In *CHI*, pages 16–23, 1986.
[18] Z. Gao, Y. Song, S. Liu, H. Wang, H. Wei, Y. Chen, and W. Cui. Tracking and connecting topics via incremental hierarchical dirichlet processes. In *ICDM*, pages 1056–1061, 2011.
[19] S. Havre, E. G. Hetzler, P. Whitney, and L. T. Nowell. Themeriver: visualizing thematic changes in large document collections. *IEEE TVCG*, 8(1):9–20, 2002.
[20] X. He and Y. Zhao. Fast model selection based speaker adaptation for nonnative speech. *IEEE TSAP*, 11(4):298–307, 2003.
[21] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE TVCG*, 19(12):2446–2455, 2013.

[22] M. Krstajić, M. Najm-Araghi, F. Mansmann, and D. A. Keim. Story tracker: Incremental visual text analytics of news story development. *Information Visualization*, 12(3-4):308–323, 2013.

[23] J. Leskovec, L. Backstrom, and J. M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[24] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, pages 1–21, 2014.

[25] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *CIKM*, pages 543–552, 2009.

[26] S. Liu, M. X. Zhou, S. Pan, Y. Song, W. Qian, W. Cai, and X. Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM TIST*, 3(2):25:1–25:28, 2012.

[27] X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from keywords. In *KDD*, pages 1433–1441, 2012.

[28] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. A. Keim. Eventriver: visually exploring text collections with temporal references. *IEEE TVCG*, 18(1):93–105, 2012.

[29] M. E. McCombs and D. L. Shaw. The agenda-setting function of mass media. *Public opinion quarterly*, 36(2):176–187, 1972.

[30] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[31] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE TSMC*, 11(2):109–125, 1981.

[32] G. Sun, Y. Wu, R. Liang, and S. Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.

[33] G. Sun, Y. Wu, S. Liu, T.-Q. Peng, J. J. H. Zhu, and R. Liang. EvoRiver: Visual analysis of topic coopetition on social media. *IEEE TVCG*, 2014.

[34] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, 2014.

[35] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. In *CHI*, pages 517–520, 2006.

[36] X. Wang, S. Liu, Y. Song, and B. Guo. Mining evolutionary multi-branch trees from text streams. In *KDD*, pages 1433–1441, 2013.

[37] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, pages 784–793, 2007.

[38] X. Wang, K. Zhang, X. M. Jin, and D. Shen. Mining common topics from multiple asynchronous text streams. In *WSDM*, pages 192–201, 2009.

[39] K. Wongsuphasawat and D. Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE TVCG*, 18(12):2659–2668, 2012.

[40] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. LifeFlow: visualizing an overview of event sequences. In *CHI*, pages 1747–1756, 2011.

[41] D. D. Woods. Visual momentum: a concept to improve the cognitive coupling of person and computer. *International Journal of Man-Machine Studies*, 21(3):229–244, 1984.

[42] P. Xu, Y. Wu, E. Wei, T.-Q. Peng, S. Liu, J. J. H. Zhu, and H. Qu. Visual analysis of topic competition on social media. *IEEE TVCG*, 19(12):2012–2021, 2013.

[43] T. Xu, Z. Zhang, P. S. Yu, and B. Long. Evolutionary clustering by hierarchical Dirichlet process with hidden Markov state. In *ICDM*, pages 658–667, 2008.

[44] D. Zhang, C.Zhai, J. Han, A. Srivastava, and N. Oza. Topic modeling for olap on multidimensional text databases: topic cube and its applications. *Statistical Analysis and Data Mining*, 2(5–6):378–395, 2009.

[45] J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora. In *KDD*, pages 1079–1088, 2010.

[46] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-R. Lin, and C. Collins. #fluxflow: Visual analysis of anomalous information spreading on social media. *IEEE TVCG*, 20(12):1773–1782, Dec 2014.

Shixia Liu is an associate professor at Tsinghua University. Her research interests include visual text analytics, visual social analytics, and text mining. She worked as a research staff member at IBM China Research Lab and a lead researcher at Microsoft Research Asia. She received a B.S. and M.S. from Harbin Institute of Technology, a Ph.D. from Tsinghua University. She is an associate editor of IEEE TVCG.
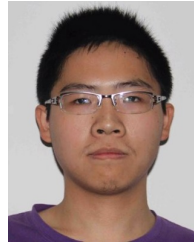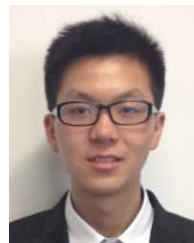
Jialun Yin is a PhD candidate in the Department of Computer Science and Technology at Tsinghua University, China. His research interests include visual text analytics and data mining. He received a BS degree in Computer Science from Tsinghua University.

Xiting Wang is a PhD candidate in the Institute for Advanced Study at Tsinghua University, China. Her research interests include visual text analytics and text mining. She received a BS degree in Electronics Engineering from Tsinghua University.

Weiwei Cui is a researcher in the Internet Graphics Group at Microsoft Research Asia. His research interests include visualization and visual analytics, with emphasis on text and graph data. He received a PhD in computer science from Hong Kong University of Science and Technology.

Kelei Cao is an undergraduate in the Department of Computer Science and Technology at Tsinghua University, China. His research interests include visual text analytics.
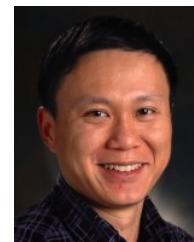
Jian Pei is currently the Canada Research Chair (Tier 1) in Big Data Science and a professor at the School of Computing Science and the Department of Statistics and Actuarial Science at Simon Fraser University, Canada. He received his Ph.D. degree at the same school in 2002 under Dr. Jiawei Han's supervision. His research interests are to develop effective and efficient data analysis techniques for novel data intensive applications. He has published prolifically and is one of the top cited authors in data mining. He received a series of prestigious awards. He is also active in providing consulting service to industry and transferring the research outcome in his group to industry and applications. He is an editor of several esteemed journals in his areas and a passionate organizer of the premier academic conferences defining the frontiers of the areas. He is an IEEE Fellow.