TextFlow: Towards Better Understanding of Evolving Topics in Text

Weiwei Cui, Shixia Liu, *Member, IEEE*, Li Tan, Conglei Shi, Yangqiu Song, *Member, IEEE*,
Zekai J. Gao, Xin Tong, and Huamin Qu, *Member, IEEE*

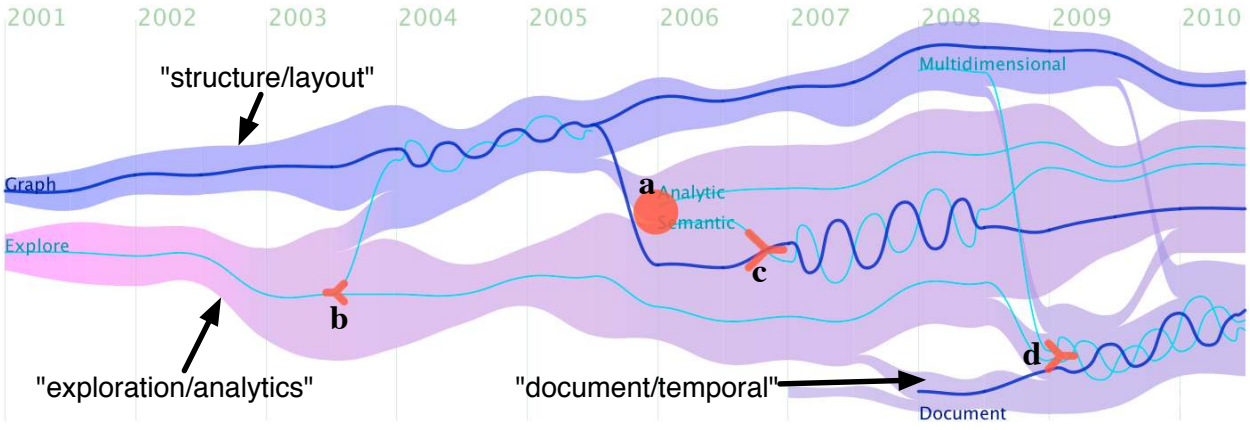

Fig. 1. Selected topic flows of VisWeek publication data with thread weaving patterns related to primary keywords "graph" and "document" (All keywords overlaid on the threads are manually labeled).

Abstract—Understanding how topics evolve in text data is an important and challenging task. Although much work has been devoted to topic analysis, the study of topic evolution has largely been limited to individual topics. In this paper, we introduce TextFlow, a seamless integration of visualization and topic mining techniques, for analyzing various evolution patterns that emerge from multiple topics. We first extend an existing analysis technique to extract three-level features: the topic evolution trend, the critical event, and the keyword correlation. Then a coherent visualization that consists of three new visual components is designed to convey complex relationships between them. Through interaction, the topic mining model and visualization can communicate with each other to help users refine the analysis result and gain insights into the data progressively. Finally, two case studies are conducted to demonstrate the effectiveness and usefulness of TextFlow in helping users understand the major topic evolution patterns in time-varying text data.

Index Terms—Text visualization, Topic evolution, Hierarchical Dirichlet process, Critical event.

1 Introduction

Understanding topic evolution in large text collections is important to many people, such as politicians, business professionals, and scholars. First, it can help them keep abreast of hot, new, and intertwining topics in their related fields, over time. Second, they could quickly gain insight into the latent topics, so that they can make proper judgements and take further actions. However, analyzing how and why topics evolve over time is not easy. Users not only need to extensively examine and differentiate individual topics, but also identify the critical events and their causes, including how new topics come into being (topic birth), what triggers and contributes to their development or disappearance (topic death), and how they gradually disintegrate (topic splitting) or dissolve into other topics (topic merging).

Much work has been devoted to effectively analyzing topic evolution. In the text mining field, researchers have developed various dynamic topic mining methods like evolutionary clustering [5, 35] and dynamic Latent Dirichlet Allocation (LDA) [28], facilitating users in analyzing the evolution of individual topics over time. On the other hand, researchers from the visualization community have designed a number of topic visualization techniques [9, 16, 17, 18] to visually illustrate the evolution of a set of independent topics. While dynamic topic mining and visualization has received much attention, little work has focused on studying topic merging and splitting patterns. Moreover, this problem has barely been touched by using visual analysis techniques to interactively analyze complex topic evolution from multiple perspectives.

There are two technical challenges that we believe are critical to visually analyze the development and change of topics over time. The first challenge is how to model the topic evolution patterns, as well as extract the critical events and the keyword correlations to provide the related information. Topics not only emerge, develop, and decline, but also influence each other by splitting and merging. Thus, it is hard to model them by using existing dynamic topic mining approaches [35] which target solely at modeling individual topics and their changes over time. The second challenge is how to visually convey and interact with the topic evolution results at different granularities to facilitate decision making. When examining the evolving topics, users may not only want to have a big picture of the global topic evolution trend, but also understand the major reasons that trigger these evolution patterns. It is therefore preferred to design a visualization that can illustrate the topic evolution results from the global evolution structure to the local salient features, such as keyword co-occurrence over time, as well as allow users to interactively explore the complex relationships between them. Furthermore, an iterative and progressive topic analysis is also

- *W. Cui, C. Shi, and H. Qu are with the Hong Kong University of Science and Technology, E-mail: {weiwei|clshi|huamin}@cse.ust.hk. W. Cui performed most of this work while at Microsoft Research Asia.*
- *S. Liu, L. Tan, Y. Song, and X. Tong are with Microsoft Research Asia. E-mail: {shliu|lit|yangqiu.song|xtong}@microsoft.com.*
- *Z. J. Gao is with Zhejiang University and performed this work while at Microsoft Research Asia. E-mail: jacobgao@gmail.com.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

expected to enable a bidirectional communication between the topic mining and visualization modules, compensating for the deficiencies in the topic mining techniques.

To tackle these challenges, we have developed TextFlow, an interactive visual analysis tool that helps users analyze how and why the correlated topics change over time. We present a topic mining model to track and connect topics over time. It starts by learning the topic merging/splitting patterns with an incremental hierarchical Dirichlet process (HDP) [23]. Then the critical events and keyword correlations are extracted to encode the local content. The output of the topic evolution analysis contains multiple-level results, including a set of topics with splitting/merging relationships to one another, a set of critical events, and the keyword correlations. To convey such complex results, we present a visualization technique inspired by river flows (see Fig. 4). Our visualization consists of three visual primitives, including topic flows, glyphs, and threads, to encode the three-level results, respectively. To produce a useful and visually pleasing layout, we formulate it as a three-level Directed Acyclic Graph (DAG), and solve it by using a force-directed simulation. TextFlow also tightly integrates interactive visualization with topic modeling techniques to facilitate users to discover the evolution patterns at different levels of detail.

To the best of our knowledge, our work is the first to help users visually analyze topic evolution patterns, as well as to allow users to interactively reason about the complex evolution relationships over time. Our work presents three technical contributions.

- **A topic mining method** that extracts a set of topics from a text collection and models the evolution relationships among them. To further assist users in understanding the underlying causes of evolutions, it also identifies the critical events and related keyword correlations to enable a deeper investigation.
- **A river-flow-based visual metaphor** that quickly conveys the salient aspects of the topic flow, while also maintaining enough context to help users examine and understand why the evolution patterns are triggered.
- **An iterative and progressive visual analysis** that enables a smooth communication between visualization and the topic mining model by interaction.

Following a review of the related work, we describe in detail TextFlow's key features, including the topic data extraction and transformation, visualization design, and interactive analysis. To evaluate the proposed method, we apply TextFlow to several text corpora. Two case studies show the promise of the work.

2 Related Work

In this section, we briefly review two categories of work related to ours, including topic evolution mining and topic trend visualization.

2.1 Topic Evolution Mining

In the field of text mining, many algorithms have been proposed to analyze topic evolution over time [35]. Existing work that is relevant to our topic mining method falls into three categories: topic detection and tracking, evolutionary clustering, and dynamic topic models.

Topic detection and tracking [1, 33] aims to automatically detect new topics and find new stories on already known topics in temporally-ordered news streams. Traditional topic detection and tracking methods use heuristic rules to detect and track topics [1]. Zhang et al. [33] first unified both detection and tracking in a probabilistic model.

Evolutionary clustering [5, 6, 31, 32, 34] focuses the problem of processing temporal data to produce a sequence of clusters. Each cluster in the sequence is similar to the cluster at the previous time step, and should accurately reflect the data arriving at that time step. The major objective of evolutionary clustering is to preserve the smoothness of clustering results over time, while fitting the data of each epoch. When applied to text data, the evolving clusters are used to track topics. Recently, evolutionary HDP [35] has been proposed to handle multiple text sources, which aims at analyzing the topic evolution patterns among multiple correlated time-varying corpora.

A dynamic topic model like dynamic LDA focuses on extending LDA to handle dynamic topics [2, 28, 27]. They allow the documents to be streaming or time-stamped. LDA-based methods mathematically model the topic generation process more elegantly. However, they introduce more parameters to be estimated [3].

Compared with the above methods, our topic analysis work not only detects and tracks individual topics over time, but also extracts the connection between topics, which is depicted as splitting and merging relationships. Moreover, we provide rich interactions to allow users to examine and analyze what triggers and contributes to topic birth, death, splitting or merging at different granularities.

2.2 Visual Topic Evolution

In the field of interactive visualization, our work is directly related to research on time-based, visual text summarization that can help users quickly discover what is inside text data. Researchers have developed a number of visualizations to help users understand the thematic variations over time within a large collection of documents.

The most popular strategy to visualize topic evolution is based on stacked graphs [4, 9, 16, 17, 29]. In the stacked graph, each layer represents a topic. The directed flow from left to right indicates the topic evolution over time. Variations in the width of each layer represents variations in strength or degree of representation. With such visualization, users can easily track the strength evolution of the topics along time. However, it fails to present the detailed content evolution to users to facilitate his/her information understanding and decision making. Recently, TIARA [18, 30] was proposed to tightly integrate interactive visualization with text summarization techniques to visually summarize a large text corpus. It allows users to examine and analyze the topic content and content evolution over time.

Themail for tracing the content evolution of the correspondence between two persons is described in [24]. In this toolkit, the content evolution is visually encoded by a set of keyword lists at different time points. In addition, several approaches have been suggested for representing content changes using tag frequencies [7, 8, 10, 14].

Compared with the above visualization research on topic evolution, TextFlow visualization targets at visually analyzing the merging and splitting relationships between evolving topics. To help users understand these relationships, a river-flow-based visualization is adopted to help users understand and analyze the evolution patterns of the examined topics. Furthermore, multiple visual cues, such as glyphs (encode critical events) and keyword threads (encode keyword correlations), are provided to differentiate independent topics, and correlate dependent topics.

Another related work is story evolution proposed by Rose et al. [20], which shows the story change and evolution over time. In their work, they target at conveying how a document changes its theme as the story develops and changes over multiple days. By contrast, our work aims at visually analyzing the topic merging and splitting relationships between topics over time. To help users understand what brings about such evolving patterns and easily trace the topic flow, multiple visual cues, such as glyphs and keyword threads, are adopted to visually illustrate various topic evolving results and enable a deeper investigation.

To help users analyze, communicate, and plan the energy distribution in a city, Riehmann et al. [19] develop an interactive Sankey diagram. Some interaction techniques are developed to facilitate the exploration and understanding of the complex flows at different levels of details. While this work inspires our visual design, we have augmented the flow metaphor to convey much richer and far more complex topic merging and splitting results. Not only does TextFlow illustrate the topic flow and their merging/splitting relationships over time, but it also depicts what triggers and contributes to such evolution patterns. As a result, TextFlow can easily facilitate the understanding of complex topic analysis results from multiple perspectives.

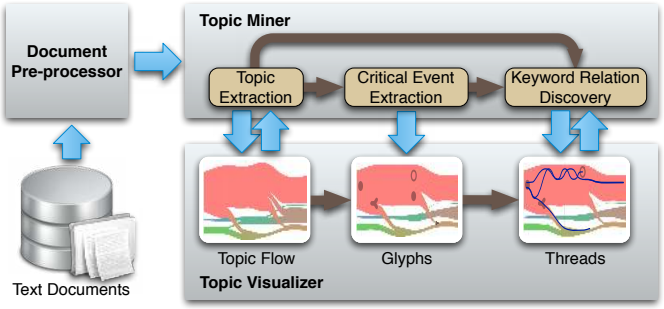

Fig. 2. TextFlow overview: system consisting of three major components: topic miner, topic visualizer, and document pre-processor.

3 System Overview

The goal of TextFlow is to help users analyze how and why topics develop and change over time. Thus, we focus our attention primarily upon enabling users to interactively consume various topic mining results, including overall topic evolution trend, critical events, and keyword correlations. Addressing these tasks requires the ability to model, navigate through, analyze, and reason about the evolving topic flow from multiple perspectives. Based on these requirements, we design and develop TextFlow.

Fig. 2 provides an overview of our system with its three main components: document pre-processor, topic miner, and topic visualizer. The input to TextFlow is a collection of text documents. The document pre-processor is tasked to extract the main document body and various metadata. Taking academic publications as an example, the metadata includes the author(s), the journal or conference name, keywords, and time stamp for each paper. The output of the document pre-processor is a collection of "clean" text content with associated metadata. The output is then sent to the topic miner which automatically extracts a set of topics along with merging/splitting relationships, as well as the critical events and keyword correlations. Given the results produced by the topic miner, the topic visualizer transforms the abstract topic mining results into a comprehensible visualization, which consists of three visual components. Specifically, a topic flow is presented to encode the overall topic evolution trend; a set of well-designed glyphs are used to represent the critical events; and keyword threads are designed to visually explain the keyword correlations. Users can then interact with the evolving topics through the generated visualization.

4 Data Processing

In this section, we first present a probabilistic model for extracting topics from a document collection and modeling the splitting/merging relationships between topics. We propose an incremental Gibbs sampling procedure based on HDP mixture of multinomial model [23], since it is effective in modeling multiple text corpora (i.e., text from different sources) and automatically determining the optimal topic numbers [23, 33]. With the incremental HDP model, we first deduce the splitting/merging behaviors between topics. Then we identify critical events to help users examine major changes inside each topic. Next, we extract keyword correlations for users to analyze topics at a finer granularity and to understand why they split and merge. Finally, we perform some post-processing such as topic and keyword ranking to allow users to examine the most salient content first.

Before going into detail, we introduce some notations, which are used in this section. We assume the data is coming in an incremental batch-mode manner, i.e., there are multiple documents coming at each time point (e.g., a month). Data can come from multiple related corpora, such as publications from different conferences. We use t to denote the time point. x^t_{ji} is the ith document in the jth corpus, represented by a vector of word frequencies. The associated cluster indicator is denoted by z^t_{ji} for the ith document in the jth corpus.

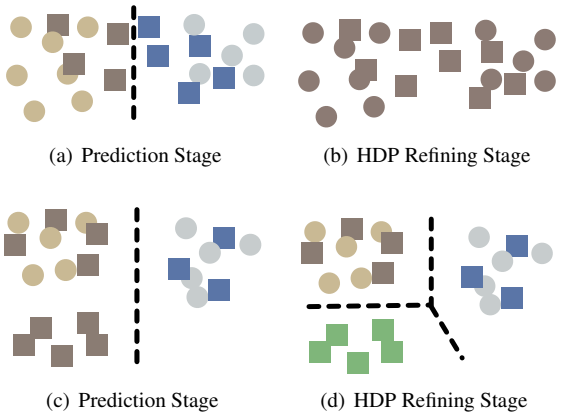

Fig. 3. An example of splitting/merging of clusters: circles representing samples at time $t-1$; rectangles encoding samples at time t.

4.1 Topic Data and Relationship Extraction

Topics are represented by document clusters. In topic extraction step, not only topics themselves but also their connections are automatically discovered. To achieve this, we first incrementally assign and refine cluster label z^t_{ji} of x^t_{ji} at each time t, by adopting two computational stages, a "prediction stage" and an "HDP refining stage". Then splitting/merging relationships among topics are computed based on document label changes.

In the "prediction stage", a classifier is used to predict the cluster label of x^t_{ji} (denoted as $z^{t,old}_{ji}$) at time t based on the previous HDP model. As shown in Figs. 3(a) and 3(c), before time t, we have some samples and an HDP model. After this stage, the labels of new incoming data at time t are predicted based on the previous HDP model.

In the next stage, the "HDP refining stage", we fix the historical labels of documents and update the label of x^t_{ji} (denoted as $z^{t,new}_{ji}$). This is a semi-supervised clustering algorithm, which both updates the data label and the HDP model [23]. Due to the property that HDP can determine an adaptive number of clusters, some topics will disappear, while some will emerge. Moreover, we can also compute the connections between dynamic topics. As shown in Fig. 3(b), the clusters at adjacent time points merge into one cluster. In Fig. 3(d), the left cluster splits into two clusters from time $t-1$ to t.

After the two computation stages, we compute the topic merging and splitting relationships by the following two types of statistics, in terms of "merging input from time $t-1$ to t" and "splitting output from $t-1$ to t". Intuitively, the first one measures how many documents in a cluster at time t are coming from different clusters at previous time points based on the previous HDP model. While the second one measures how many documents are sampled into different clusters for a specific cluster. We compute them respectively as follows.

Merging input from time $t-1$ to t: the proportion of cluster r coming from cluster s is measured by the difference between $z^{t,old}_{ji}$ and $z^{t,new}_{ji}$ at current time t:

$$P^{in}_t(s \to r) \triangleq \frac{\sum^t_{\tau=t-T_{win}+1} \sum_{i,j} I(z^{t,old}_{ji} = s \;\&\; z^{t,new}_{ji} = r)}{\sum^t_{\tau=t-T_{win}+1} \sum^{n^t}_{i=1} I(z^{t,new}_{ji} = r)} \quad (1)$$

where $I(\cdot)$ is an indicator function that flips between binary values, i.e., $I(true) = 1$ and $I(false) = 0$. Here we consider the probability based on both the data at time t and the historical data in a time window T_{win}.

Splitting output from time $t-1$ to t: the proportion of cluster s flowing into cluster r is measured by the difference between $z^{t,old}_{ji}$ and

$z_{ji}^{t,new}$ from time $t-T_{win}+1$ to t:

$$P_{t-1}^{out}(s \to r) \triangleq \frac{\sum_{\tau=t-T_{win}+1}^{t} \sum_{j,i} I(z_{ji}^{\tau,old} = s \;\&\; z_{ji}^{\tau,new} = r)}{\sum_{\tau=t-T_{win}+1}^{t} \sum_{j,i} I(z_{ji}^{\tau,old} = s)}. \quad (2)$$

4.2 Critical Event Extraction

Two basic critical events include cluster (or sub-cluster) birth and death. The birth (or death) of a cluster denotes an emerging (or disappearing) topic (or sub-topic) in the text stream. They are detected by applying a heuristic rule. In our implementation, we maintain a hash table of the clusters for each time point, and the critical events of birth and death can be easily detected by comparing the hash tables between adjacent epochs. The importance of critical events is computed based on the number of documents in that cluster at the corresponding time point.

Another non-trivial type of critical event is topic splitting/merging. Although all the splitting/merging points are already denoted by the flow shape, they are not equally important. Critical ones represent severe content changes, instead of slowly evolving. To extract this type of critical event, we first rank the splitting/merging events by using both the number of branches at the points and the entropy of the splitting/merging proportions. Then we select the ones with the highest ranking scores as the critical events.

Mathematically, the score for a merging event is formulated as:

$$R(r,t) = |\mathcal{N}_r| \cdot H_t(r) = |\mathcal{N}_r| \cdot \kappa_B \sum_{s \in \mathcal{N}_r} -P_t^{in}(s \to r) \ln P_t^{in}(s \to r) \quad (3)$$

where $R(r,t)$ is the ranking score of cluster r at time t, $H_t(\cdot)$ is the entropy score at t, and κ_B is the Boltzmann constant. $\mathcal{N}_r$ is the neighborhood set of cluster r. It consists of the branch clusters that flow into r, and $|\mathcal{N}_r|$ is the number of elements in $\mathcal{N}_r$. Similarly, the ranking score of a splitting event is defined as:

$$R(s,t) = |\mathcal{N}_s| \cdot H_t(r) = |\mathcal{N}_s| \cdot \kappa_B \sum_{r \in \mathcal{N}_s} -P_{t-1}^{out}(s \to r) \ln P_{t-1}^{out}(s \to r) \quad (4)$$

where $R(s,t)$ is the ranking score of cluster s at time t. $\mathcal{N}_s$ is the neighborhood set of cluster s; its elements are the branch clusters that flow out of s. $|\mathcal{N}_s|$ is the number of elements in $\mathcal{N}_s$.

Since our data model uses keywords to summarize topics, the critical events likely involve intense keyword changes, including keyword birth, death, and splitting and merging among topics. Finding critical events related to the changed keywords may help users understand the content evolutions.

4.3 Keyword Correlation Discovery

To help users better understand the major reasons triggering topic evolutions, we also extract "noun phrases," "verb phrases", and "named entities" in each document, and count co-occurrences among them. We provide a list of top syntactic and semantic keywords as well as a list of top keyword pairs to take users to a detailed analysis.

4.4 Topic Keyword Ranking

Previous study on keyword ranking methods [22, 30] has shown that the following two criteria are very useful in selecting the interesting keywords to represent the topic content at each time point. First, the keywords at each time should reflect distinctive content, thus we could see the evolving of the topic (distinctiveness). Second, the keyword sets along time together will cover the total content of the topic (completeness). In our work, we follow these two criteria and slightly modify them to adapt to our incremental batch-mode manner. The rank of a keyword w in cluster k at time t is given by:

$$\text{Weight}(w)_k^t = \frac{\text{TF}(w)_k^t}{\sum_k \text{TF}(w)_k^t} \cdot \exp\left(-\lambda \cdot \text{Weight}(w)_k^{t-1}\right) \quad (5)$$

where w represents a word, $\text{TF}(w)_k^t$ is the term frequency of w in cluster k at time t, $\sum_k \text{TF}(w)_k^t$ is the sum of $\text{TF}(w)_k^t$ among different clusters, $\text{Weight}(w)_k^{t-1}$ is the weight computed for time $t-1$. Note that $\exp\left(-\lambda \cdot \text{Weight}(w)_k^{t-1}\right)$ can be regarded as a decay factor of each keyword. If a keyword appears at the last time point with very high score, it will be ranked lower at this time point. Contrarily, if it has not been shown before, it should be emphasized. The coefficient λ is a parameter controlling the velocity of the decay, which is set to 0.9 in our system.

5 Visual Design

Although the mining results produced by the topic miner well summarize the topic evolutions in the input documents, they are abstract and difficult to understand. We thus developed a Texflow visualizer for presenting the three-level mining results comprehensively in a single view. Specifically, a topic flow graph inspired by the river metaphor is presented for visualizing the overall topic evolution patterns (Section 5.1), a set of glyphs is applied over the flow for encoding the critical events (Section 5.2), and a set of keyword threads is designed to reveal the keyword correlation details in each topic evolution flow (Section 5.3). Combined with a tag cloud view and a timeline view (Section 5.4), the Texflow system provides an efficient solution to various visual exploration and analysis tasks.

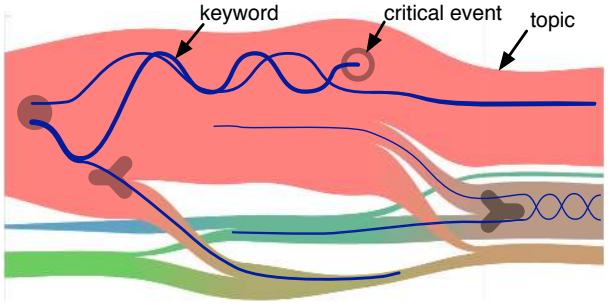

Fig. 4. An example of TextFlow visualization, which consists of four topic flows, four critical events, and five keyword threads.

5.1 Topic Evolution as Flow

A naive solution to visualizing the topic evolution is to display all topic evolutions as a stacked graph. Although this approach can well illustrate the linear topic evolution along time [4, 9], it cannot well present complicated topic evolution patterns that include topic merging and splitting. To tackle this issue, we adopt a river flow metaphor (see Fig. 4) for visually conveying the topic evolution patterns in Textflow.

As shown in Fig. 4, we represent the evolution of topics along time as a topic flow graph. The varying flow height along the x-axis encodes the number of the documents that belong to this topic at each time point. Like a river flow in the real world, the topic flow can be either split into several branches when the corresponding topic splits, or merged with several other branches into one flow when their corresponding topics merge into one topic. After splitting (or merging), the branch with the content that is most similar to the topic before splitting (or after merging) becomes the main branch, with the rest as the secondary branches (Fig. 5). The heights of all branches are determined by the number of documents in the corresponding branch topics.

To visualize evolution patterns, we stack the topic flows of all topics vertically and align them based on their time stamps. To determine the order of the topic flows in the stack, we develop a flow layout algorithm, which can reduce the visual clutter caused by flow branch crossings and generate a smooth flow outline. The details about our flow layout algorithm are described in Section 6. To help users easily identify and track the specific topic flow(s), we uniformly pick colors from a rainbow color spectrum for original topic flows according to its vertical order in the stack. For the derived topics generated by merging and splitting, we adopt the following coloring strategy. The color of the merged topic flow is the blending of the colors of the corresponding branches. The blending weights are determined by the ratio of branch heights to the original flow height. The color of each branch after topic

splitting is determined from the color of the original flow similarly. As a result, the color of the topic is actually influenced by the color of multiple topics in its history and thus provides a good supplementary visual cues for topic tracking. Our color scheme may cause misleading colors in some cases. For example, too many splitting/merging relationships likely lead to very similar colors among different flows. In such case, our color strategy may fail to differentiate related topics, users can then rely on the flow shape to track topics.

Our topic flow graph can be regarded as an enhancement of the traditional stacked graph. By hiding the secondary branches, the topic flow graph can be easily transformed into a stacked graph (see Fig. 5). In our implementation, we use the stacked graph as the initial visualization of the overall topic evolution patterns and then transform the visualization into a topic flow graph for visualizing the detailed topic merging and splitting patterns in which users are interested. This scheme provides an easy way for users to understand and explore the complicated topic evolution patterns and flatten their learning curve.

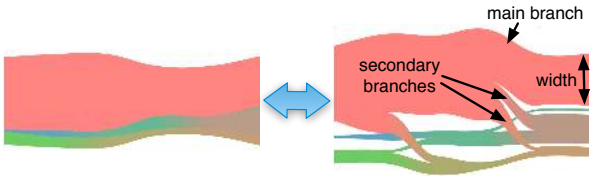

Fig. 5. The conversion between a stacked graph and a topic flow graph.

5.2 Critical Event as Glyph

Four intuitive glyphs are chosen to represent the birth, death, splitting, and merging events, respectively (see Fig. 6). We overlay these glyphs at the time points where they occur to illustrate the critical events of the corresponding topic flow. In our data model, topics are represented by a set of keywords. Therefore, those glyphs can also help users find intense keyword changes (see Fig. 4). The size of the glyphs is computed by the importance score of the critical events described in Section 4.2. The bigger a glyph is, the more important its corresponding critical event is.

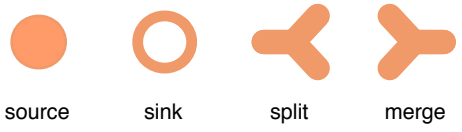

Fig. 6. The glyphs represent four different critical events, respectively.

5.3 Keyword Correlation as Thread

The keyword correlation is the third aspect that we consider. It is very important in topic analysis, because it allows users to understand why the topics develop and change at a finer granularity.

A straightforward way to visually encode keyword correlations is to overlay word clouds on the topic flow (see Fig. 7(a)), which is also probably the most common method to show a collection of keywords to users in the field of information visualization [8, 26, 25]. However, this representation is not effective and useful in our scenario. First, a single word cloud fails to show the content evolution of keyword correlations. Even if we can adopt a set of word clouds to encode the content evolution over time, this may introduce additional visual clutter to the topic flow. Second, it cannot illustrate the co-occurrence interactions between keywords over time, which characterizes the major critical events in topics.

Another option for capturing the dynamic content and co-occurrence interactions over time is to use a set of polylines to represent the keywords (Fig. 7(b)). Co-occurrence relations could be represented by distances between them. In addition, a force-based model could be adopted to lay out the polylines. However, we can not use it in our design either. The distance between the polylines encodes their closeness. On the other hand, the distance is also constrained by flow boundaries. For example, at the time point where the flow is thin, the polylines may not be close to each other. However, because of the limited space, they have to be placed very close and may give users the wrong impression (see Fig. 7 (b)).

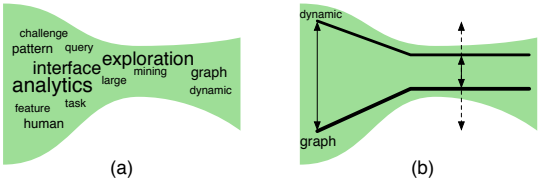

Fig. 7. Two alternative ways to represent the keyword correlation: (a) placing word clouds on the flow; (b) using polylines and their distances between polylines to encode word correlations.

Instead, we design a novel visual primitive called *thread* to solve these problems. Fig. 4 shows a simple example of thread. Intuitively, if one keyword appears in a topic at a specific time point and disappears from the topic at another time point later, we draw a thread in that topic flow between the two time points. Meanwhile, we use the weaving effects (see Fig. 8) to represent the co-occurrence interactions between two or more keywords. In particular, we select one keyword as the primary keyword (the thick line in Fig. 8). In the thread representation, we only consider the co-occurrence interactions between the primary keyword and other non-primary keywords.

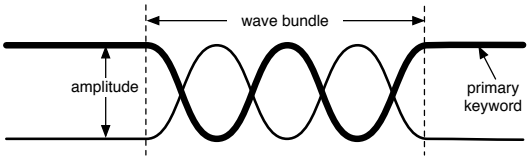

Fig. 8. The visual attributes in the thread weaving: wave bundle and amplitude.

We also adopt various visual attributes in thread weaving to encode different information:

- **Wave bundle:** If several threads are weaving together, we use the weaving bundle to encode the co-occurrence interaction at that time point. The length of the wave bundle represents the time period length of the co-occurrence.
- **Amplitude:** We use the amplitude to encode the occurrence number of all involved keywords. The bigger the weaving amplitude, the more the corresponding weaving keywords appear at that time point. Usually the height of a topic at one time point is proportional to its related keyword number at that particular time point, so the topic flow can well accommodate the thread weaving with different amplitudes.

5.4 Complemented with Tag Cloud and Timeline

Although the three visual components in the TextFlow view provide a comprehensive graphical representation for topic evolutions, they all ignore the detailed textual information in the mining results, which are important for some visual analysis tasks. To solve this issue, we added two well-accepted visual components to be synchronized with the TexFlow view: a tag cloud view placed at the bottom left of our system UI, and a timeline view placed at the bottom right (Fig. 13). These three views are coordinated with each other to help users interactively inspect and refine the mining results. Specifically, when users select some topics in the TextFlow view, the tag cloud view and the timeline view will show the words and sentences with high importance scores, in the corresponding selected topics, respectively. More details about the interactions of our system can be found in Section 7.

6 Layout Algorithm

In this section, we describe the layout algorithms for the visual elements discussed in Section 5. The position of a glyph is easily determined by its topic and time point where it occurs. As a result, here we mainly focus on the topic flow layout and thread layout.

6.1 Problem Formulation

Mathematically, the topic flow layout together with the thread layout can be formulated as a three-level Directed Acyclic Graph (DAG). As shown in Fig. 9, the first level DAG corresponds to the topic flow (Fig. 9(b)). Here nodes represent the clusters at each time point, and edges indicate the content flow between two consecutive clusters along time. The second level encodes the DAG of keyword bundles (Fig. 9(c)). In this graph, nodes represent bundles that contain highly interacted keywords at that time point. Edges represent the keyword flow. The third level represents the DAG of keyword threads, which is derived by splitting each bundle node into several thread nodes based on the number of the keyword threads it contains (see the middle of Fig. 9(d)). Here a node represents a keyword segment involved in a bundle at each time point, while an edge indicates a keyword flow from one node to another.

When laying out the DAG of keyword bundles, it has a boundary constraint that each bundle node should be placed inside its topic layer. While for the keyword thread layout, it has the constraint that each thread node should be placed inside the bundle it belongs to. Theoretically, the three-level hierarchical DAG layout can be simplified into three DAG layouts with regular or irregular boundary constraints.

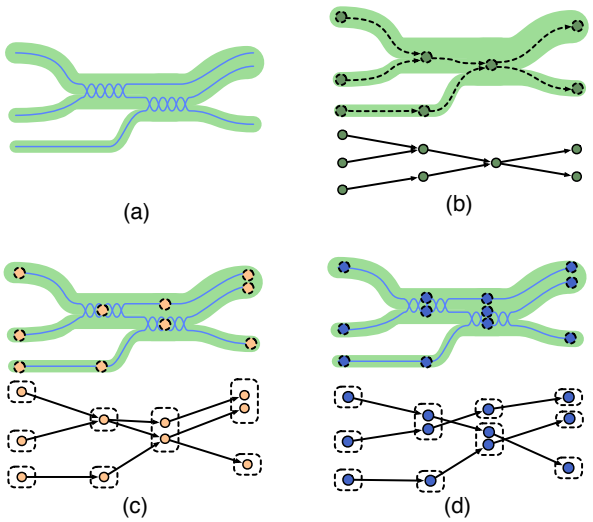

Fig. 9. The three-level model for topic flow graph layout (dotted rectangles indicating boundary constraints): (a) the original structure; (b) first level: topic flows; (c) second level: bundles; (d) third level: threads.

6.2 Algorithm

In our DAG layout, we focus on addressing two aesthetic criteria:

- Decide the optimal order of the nodes at each time point to reduce edge crossings;
- Calculate the exact position for each node to smooth topic layers (keep edges short) and favor symmetry.

Accordingly, our algorithm consists of two major steps: node ordering and node placement.

Node Ordering Too many edge crossings make the topic flow visually cluttered and thus difficult to explore. It is therefore important to order the nodes at each time point to reduce the edge crossings in the layout. However, minimizing edge crossings in a layout is an NP-complete problem [12]. To tackle this issue, we leverage a force simulation method based on median heuristics [13].

We first define some notations useful for subsequent discussions. Let $\mathscr{G}$ denote a DAG with k layers (i.e., time points in our case), and $\mathscr{L}_i$ ($1 \le i \le k$) be the ith layer. v_{ij} represents the node with jth order in layer $\mathscr{L}_i$. In addition, the incoming and outgoing nodes of v_{ij} are denoted as v_{ij}^I and v_{ij}^O, respectively.

In our layout, we define the median function of node v_{ij} as:

$$m_{ij} = \omega \cdot j + (1-\omega) \sum_{v_{kl} \in v_{ij}^I \cup v_{ij}^O} w_{(ij)(kl)} \cdot l \tag{6}$$

where the ω ($0 \le \omega \le 1$) is the control parameter, and $w_{(ij)(kl)}$ is the weight of the edge between v_{ij} and v_{kl}, encoding the ranking score of topic merging and splitting.

Our node ordering is determined by an iteration process described in [13, 15]. First, every node is given an initial vertical order at its layer. The initial order is determined by the node visiting order in depth-first search. Based on the initial order, a sequence of iterations is then performed to improve the ordering.

After the above iteration is converged, we apply a local adjustment to update the node order at each layer alternately by fixing the node order in other layers. It is developed for solving the deadlocks generated in the iteration. Unlike the traditional directed graph layout algorithms that only consider the edge crossing number, we combine the edge weight and edge crossing to evaluate the metrics of crossings. For each layer, we calculate the metric for the current order, and update it only when the metric is reduced. Such evaluation can help to avoid crossings between edges with larger weights, and thus improve the visual quality of the final layout.

Node Placement The major objectives of node placement are to improve the smoothness of individual topic flows, as well as to reduce the turning angles of splitting/merging flow branches. To achieve this, we follow a classical directed graph layout method to formulate them as a problem of minimizing the edge length.

$$\begin{aligned} \min_{\mathbf{Y}} & \sum_{v_{ij}} \sum_{v_{kl} \in v_{ij}^I \cup v_{ij}^O} (y_{kl} - y_{ij})^2 \\ s.t. \quad & y_{ij} - y_{i,j-1} \ge \tfrac{h_{ij}+h_{i-1,j}}{2} \qquad \text{if } j > 0 \\ & y_{ij} \ge b_{li} + \tfrac{h_{ij}}{2} \\ & y_{ij} \le b_{ui} - \tfrac{h_{ij}}{2} \end{aligned} \tag{7}$$

where y_{kl} and y_{ij} represent the vertical coordinates for node v_{kl} and node v_{ij}, respectively, and $Y = \{y_{ij} | v \in \mathscr{G}\}$. The h_{ij} is the size of node v_{ij}, while b_{li} and b_{ui} represent the lower and upper boundaries of the layout region.

This is a quadratic programming problem with linear constraints, which can be easily solved by a standard quadratic programming tool such as Mosek and CPLEX. However, force-directed simulations can achieve better performance [11] and support related interactions such as the local adjustment of node order. As a consequence, we decide to solve this problem by a force-directed simulation. We regard the gradient value of the object function in Eq. 7 with respect to variable y_{ij} as the force of node v_{ij} in the simulation. Unlike traditional force-directed simulations, which do not consider the separation constraints, we deal with the separation constraints in Eq. 7 by using a collision simulation. Specifically, we simulate the collision, which visually represents the overlapping of nodes, by using completely inelastic collision. When two nodes overlap, we calculate the weighted average speed from both nodes, and then set the speed of each node to the average value. Such simulation can fully support the interactive adjustment of layout while maintaining the separation constraints between nodes.

Unlike the dynamic topic model, our layout algorithm is not incremental. When the new data is coming, the entire river layout need to be recalculated, which may cause the stability issue. Therefore, the layout algorithm works better for the data that is all ready, rather than for dynamic text streams. Fig. 10 demonstrates one result generated by our layout algorithm. With our method, branch crossings are reduced and individual topic layers are smoothed.

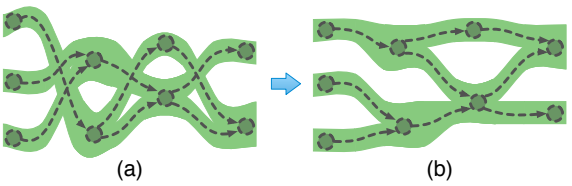

Fig. 10. One example of our topic flow layout: (a) initial input; (b) our layout result.

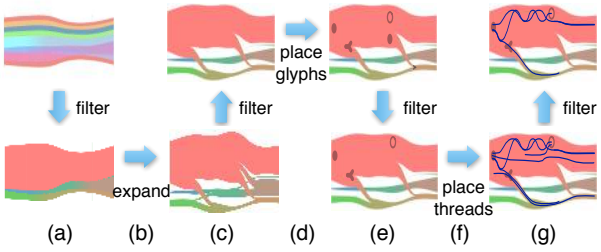

Fig. 11. The exploration pipeline: (a) select a subset of interest based on users' interest or system recommendation; (b) expand the topics to see their splitting and merging patterns; (c) remove trivial or irrelevant branches; (d) and (e) extract major critical events based on current flow patterns; (f) and (g) explore and adjust threads around the selected critical events based on users' interest or system recommendation.

7 Interactive Exploration

Fig. 11 shows a general exploration pipeline in our system. Our pipeline starts with our simplest representation: stacked graphs. During the progressive exploration, users will know more and more about the data, so that they could filter out irrelevant data step by step. Meanwhile, the visualization also gradually improves its visual complexity to reflect their insights. There are two major operations in this pipeline: interactive filtering and recommendation.

7.1 Interactive Filtering

At every step in our exploration pipeline (as shown in Figs. 11(a), (c), (e), and (g)), our system supports various interaction techniques to help users access more information in the backend datasets, so users could filter the data based on what they find. When users interact with our visualization, there are two general operations that they can choose: *hovering* and *selecting*.

Hovering Hovering on a visual element provides users with simplified information, so that they can decide to select it to see more information, or move on to other visual elements. For example, when hovering on a topic, most representative keywords, which are chosen by our data model, will be overlaid on top of the topic layer by using the sweeping line algorithm [21], and the secondary branches between this topic and the other topics are also indicated by thin lines (see Fig. 12). Similarly, when users hover the cursor on a critical event or a thread, a tooltip will pop up to show the representative keywords in the critical event and corresponding keyword for the thread, respectively. In particular, considering the meanings of our critical events (i.e., birth, death, split, and merge), our data model is tweaked to prefer keywords that reflect these meanings.

Selecting Selecting a visual element has two purposes. First, selecting a visual element could provide users with more information than hovering, all the extra information is brought to users by the linked views, such as the tag cloud view and timeline view. For example, when users click a topic in the TextFlow view, the tag cloud and the timeline will change accordingly and show more important keywords or sentences related to the selected topic. Second, selecting a visual element could make the backend model aware of users' interest, and then make proper recommendations to help users with further exploration and filtering.

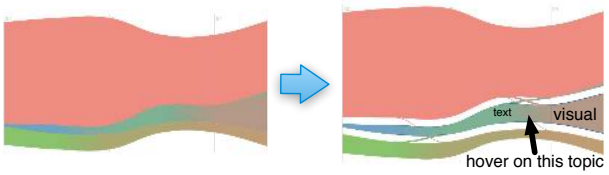

Fig. 12. Hovering on a topic: thin lines indicate the splitting/merging relationships between the hovered topic and the other topics.

7.2 Recommendation

Our mining model could help the filtering task in two ways. First, when users select a topic, our data model will calculate the correlation scores (i.e., the sum of the splitting/merging scores described in Section 4.2) between the selected topic and other individual topics. Thus, the visualization could change the opacity values of the topic flows accordingly. Therefore, if users want, they could include the highly-correlated topics into the next step of splitting/merging exploration. Second, when users select a keywords from the tag clouds, the backend data model will search for highly co-occurring keywords in the dataset. Thus, the visualization could draw their co-occurrence patterns in the view to help users find unexpected patterns in the content.

8 Applications

To demonstrate the usefulness and effectiveness of our TextFlow system, we have applied our system to two datasets: paper publications and political news. In this section, we describe several interesting findings from our explorations. All the experiments are conducted on an iMac with a 2.8GHz Intel Core i5 CPU and 4GB memory. Our system first extracted various topics in the preprocessing stage, which took several minutes, for each dataset. After the preprocessing, all layout algorithms and users interactions were performed interactively.

8.1 Publication Data

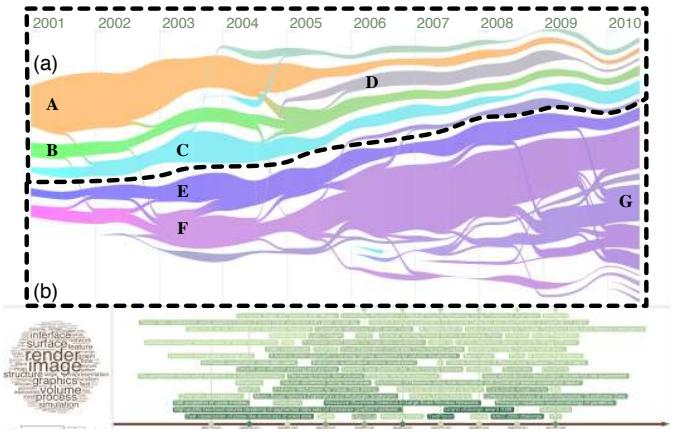

Fig. 13. Topic flows for VisWeek publication data: (a) topics related to scientific visualization; (b) topics related to information visualization.

We collected 933 research papers published in two venues: IEEE Visualization (Vis) and IEEE Information Visualization (InfoVis) from 2001 to 2010. Then we processed the paper abstracts and titles along with their meta data, such as author(s) and keywords. The vocabulary size is 9,020. Fifteen topics are learned from this paper corpus (see Fig. 13). After we use the tag cloud and the timeline views to quickly identify the content in each topic, some patterns stand out immediately. For example, all the topic flows can be generally divided into two groups. The topics (marked as **A**-**D** in Fig. 13) in the upper part (see Fig. 13(a)) are about Vis, which represent "geometry/rendering", "flow/vector", "volume/rendering", and "tensor/fiber", respectively. From the figure, we can see that the Vis topics slightly

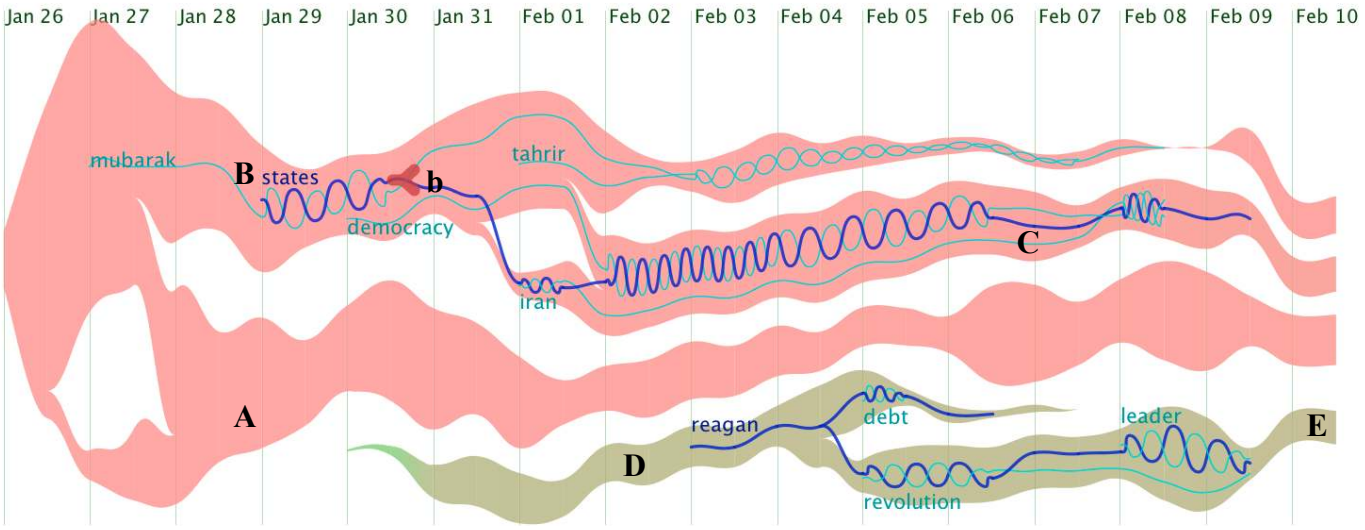

Fig. 14. Selected topic flows of Bing news data with the thread weaving patterns related to primary keywords "state" and "reagan" (All keywords overlaid on the threads are manually added).

declined in the last decade. In addition, we can clearly see there are some split/merge patterns between those flows before 2006. However, after 2006, Vis topic flows have been consistently developing independently. On the other hand, the topic evolution patterns in InfoVis (see Fig. 13(b)) are quite the opposite. There are three major flows in it (marked as **E**-**G** in Fig. 13), which represent "structure/layout", "exploration/analytics", and "document/temporal", respectively. In this part, we can see that the overall trend of InfoVis is growing, especially for topic **F** "exploration/analytics". In addition, all the topics are much closer than the topics in Vis, since we can see a lot of splitting/merging relationships among them. This may indicate that this field, unlike Vis, is still undergoing dramatic changes, and different topics are influencing each other. Furthermore, we can see that topic **F** "exploration/analytics" has much more outgoing branches than incoming branches, which suggests that this topic is a vibrant contributor to many other topics. Notably, in the bottom right corner of the view, there are several newly developed topics which are mainly derived from topic **F** "exploration/analytics".

Since we are more interested in the major topics of InfoVis, we select topic **F** "exploration/analytics" in the TextFlow view. Then we follow the system's recommendations to keep several of the most related topics for further exploration. In the filtered topic flows, we examine the content evolution between different topics. With the help of topic analysis, we further simplify the view by just showing several most important split/merge branches, shown in Fig. 1. In the figure, we find several interesting content flows. For example, there is a very distinct source in the topic **F** "exploration/analytics" in 2006 (marked as **a** in Fig. 1), which is in the middle of the topic flow. We are interested in finding out what causes it, so we click that glyph and bring out the documents related to it. It turns out that most of the papers are related to "analytics", which indicates that analytics has suddenly become a very hot word in that year. We believe that it is not a coincidence that 2006 is also the year that the first IEEE VAST Symposium was held. At that time, it still does not emerge as a new topic flow; instead, it blends in well with the old "exploration" topic. On the other hand, it also indicates that "analytics"-related research brought a big change to the old InfoVis field, otherwise it will not appear as a major source in the flow. We turn on the keyword threads starting from that source to see what this source brings to the existing topic. From the threads, we find out that the major keywords include "intelligent", "temporal", "network", and "analytics". Although these words do appear in previous InfoVis papers occasionally, they are clearly more emphasized since that year, just like "analytics".

In addition, we also find several splitting and merging critical events (marked as **b**-**d** in Fig. 1) in the display. Specifically, critical events **b** and **c** indicate that there are some major interactions between topics **E** "structure/layout" and topic **F** "exploration/analytics". By turning on the threads related to them, we explore the content flows between them. Among the displayed threads, we find three representatives: "graph", "explore", and "semantic", which are labeled in Fig. 1 manually. We can see that topics **E** and **F** focus on layout and exploration, respectively before 2003. After 2003, topic **E** starts to involve exploration in its layout process, while topic **F** starts to heavily rely on abstract data models, such as graphs and treemaps, to help analysis tasks. The third critical event **d** indicates that the topic **G** "document/temporal", which has recently become another major topic in InfoVis around 2009, is heavily influenced by topics **E** and **F**. By looking into the related threads, we find that this topic benefits from various "multidimensional"-related techniques in topic **E** and "interactive exploration" techniques in topic **F**.

8.2 Bing News Data

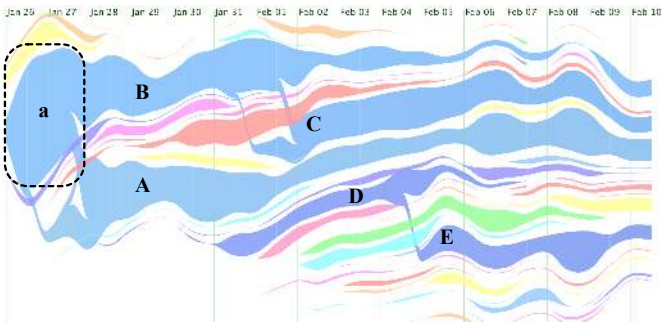

Fig. 15. An overview of topic flows for Bing news data: most topics evolve independently; three topics (marked as **A**-**C**) derived from the same cluster (marked as **a**), while topic **E** is derived from topic **D**.

This news dataset contains 2,980 articles of Bing news, which are related to "Obama" and span 16 days (from January 25 to February 10). The vocabulary size is 31,726. Fig. 15 shows the topic flow result generated by our system, which includes over 30 topics covering various content, such as "Egypt crisis", "health/insurance", "budget/finance". From the figure, we can clearly see that most of them evolve independently and rarely split or merge after the third day. Therefore, we filter out all the other simple flows, and focus on

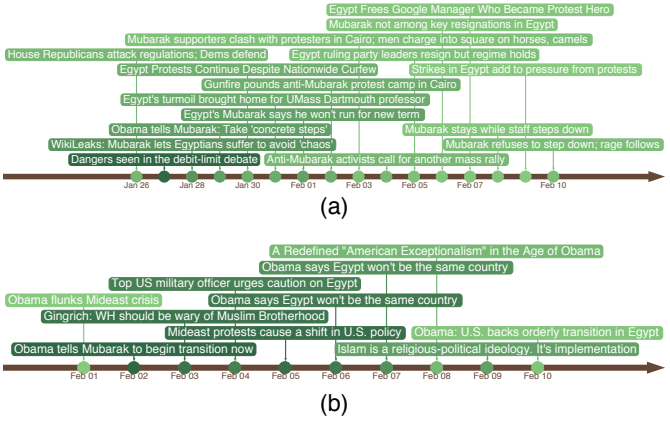

Fig. 16. Comparing timelines extracted from different topics: (a) timeline extracted from topic **B**, which mainly describes protest events happening in Egypt; (b) timeline extracted from topic **C**, which focuses on the Obama's government's actions on events in Egypt.

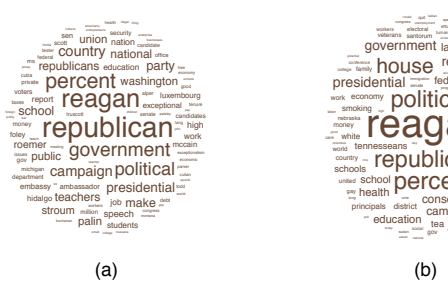

Fig. 17. Comparing word clouds extracted from different topics: (a) word cloud extracted from topic **D**; (b) word cloud extracted from topic **E**.

these major split/merge patterns. Among all the remaining topics (see Fig. 14), we find one topic related to "budget/spending" (marked as **A** in Fig. 14), two topics related to "protest in Egypt" (marked as **B** and **C** in Fig. 14), and two topics related to "republican/campaign" (marked as **D** and **E** in Fig. 14). To understand what causes those topics to split, we explore the content close to the splitting/merging points.

At the first time stamp, we find that the dominant cluster **a** (Fig. 15) is very general, and contains various content, such as "budget/spending" (topic **A**) and "protest in Egypt"(topic **B**). The variety of topics in cluster **a** is due to the limitation of our data mining model. Without sufficient data samples, unrelated topics could mix together and lead to clusters making no sense. As the data is accumulated, cluster **a** finally splits into two completely separate topics, **A** and **B**.

Topics **B** and **C** are both related to "protest in Egypt". However, we still discover a subtle difference between them by reading the timelines extracted from them (see Fig. 16). In the figure, we can see that the news articles in topic **B** mainly describe the events happening in Egypt during the protest, while the articles in topic **C** mainly discuss the Obama government's actions on the events. To find more details about the splitting, we select the critical event **b** (see Fig. 14), and navigate through news article titles related to it. One of them is "U.S. Officials Keep Close Eye on Egyptian Upheaval", which indicates that the situation became worse, so that topic **C** grows bigger inside topic **B** and finally separates from it. In addition, we turn on the keyword threads around the critical event **b** (see Fig. 14). Among those threads, we find that several keywords (e.g., "states" and "white house") that previously exist in topic **B** directly disappear and go into topic **C** (see Fig. 14). This pattern also proves that the focus splits between topics **B** and **C**. When the keyword "states" flows into topic **C**, we also find it has several correlations with several interesting words, such as "Iraq", "Iran", "Yemen", and "democracy". In particular, we find "democracy" co-occurs with "states" most of the time, and "Iran" co-occurs with "states" on and off (see Fig. 14). On the other hand, all the threads remaining in topic **B** are related to the events that happened in Egypt, such as "military", "tahrir", and "violence". In particular, we find "tahrir" is a new keyword that appears around February 1 and co-occurs with "mubarak" ever since (see Fig. 14). By looking into the documents related to both words, we find that "tahrir square" is the major location protesters rally to ask "mubarak" to step down.

Topics **D** and **E** both contain political party events that are highly related to republicans and cover various contents, such as "education", "job", and "finance". By comparing word clouds generated from both topics, we find that keyword "reagan" in topic **E** is much larger than the one in topic **D** (see Fig. 17). We click this word to see its distributions in the topic flows (see Fig. 14), and find out that "reagan" appears right before the splitting point and it is split into both topics. By exploring the origin of "reagan", we find out there are several documents talking about the upcoming events for celebrating the 100th birthday of "Ronald Reagan". Therefore, topic **E**, which is directly derived from topic **D**, should mainly focus on the celebration. In topic **E**, we find several interesting keywords highly correlated to "reagan", such as "tax", "job", and "optimism", which may indicate several aspects of the former president that those news articles describe. Based on these keywords, we find people generally give positive comments to him and consider him as a capable leader. Fig. 14 shows that people discuss the Reagan revolution first and leadership later.

9 Discussion and Future Work

In this paper, we have presented TextFlow, a visual topic analysis system to help users explore and understand topic evolutions. Compared with previous solutions, our system is unique in two ways. First, it allows users to reason about various topic evolution patterns at different granularities. Second, it enables seamless communication between topic mining and interactive visualization, supporting an iterative "refine and analyze" analysis cycle. Through two case studies, we have demonstrated the usability of our techniques in facilitating users to visually analyze the topic evolution patterns in a document collection.

Our design also has some limitations. First of all, meaningful threads are not easy to choose. If we provide too few keywords, the user may have little to go on. On the other hand, if we provide too many keywords, the visualization is cluttered and hardly anything can be inferred from it. As a consequence, users have to explore thoroughly to find a few keywords that best describe the content evolutions within topics. To assist users with this, we design several exploratory interactions. In addition, we provide two complementary views: timeline view and tag cloud view. The timeline view, which shows meaningful snippets, can provide a more informative overview of content over time, while the tag cloud view can provide possible candidates for users to select threads. By interacting with them, users may gain more and more insight into the dataset, and find the appropriate threads that reveal underneath patterns. Second, it is not easy to perform comparison task in our current system. For example, to compare content between different topics, users have to select them back and forth to see the changes in the timeline view and tag cloud view. To help users with this task, one straightforward way is to show the contents (i.e., timeline and tag cloud) of selected topics in separated windows, so that the side-by-side comparison is possible.

In the future, we are interested in dealing with the scaling problem to enable TextFlow to cope with billions of documents both from the mining side and visualization side. We would also like to study how to visually compare different topics effectively. In addition, we plan to use various application specific features to extract more accurate topic merging and splitting patterns.

Acknowledgments

The authors would like to thank Stephen Lin for proofreading the paper, Kwan-Liu Ma for his support, and the anonymous reviewers for their valuable comments. This work was supported in part by grant HK RGC GRF 619309 and HKUST Overseas Research Awards.

REFERENCES

[1] J. Allan, editor. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[2] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[4] L. Byron and M. Wattenberg. Stacked Graphs–Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008.

[5] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12nd ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 554–560, New York, NY, USA, 2006. ACM.

[6] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 153–162, New York, NY, USA, 2007. ACM.

[7] C. Collins, F. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 91–98. IEEE, 2009.

[8] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu. Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, 30(6):42–53, 2010.

[9] M. Dörk, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

[10] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 193–202, New York, NY, USA, 2006. ACM.

[11] T. Dwyer, Y. Koren, and K. Marriott. IPSep-CoLa: An incremental procedure for separation constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):821–828, 2006.

[12] P. Eades, B. McKay, and N. Wormald. On an edge crossing problem. In *Proceedings on 9th Australian Computer Science Conference*, pages 327–334, Australian National University, 1986.

[13] P. Eades and N. Wormald. The median heuristic for drawing 2-layers networks. In *Technical Report 69, Dept. of Computer Science, Univ. of Queensland*, 1986.

[14] D. Fisher, A. Hoff, G. G. Robertson, and M. Hurst. Narratives: A visualization to track narrative events as they develop. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 115–122, 2008.

[15] E. R. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo. A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230, 1993.

[16] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[17] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[18] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM.

[19] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive sankey diagrams. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 31–39, Washington, DC, USA, 2005. IEEE Computer Society.

[20] S. Rose, S. Butner, W. Cowley, M. Gregory, and J. Walker. Describing story evolution from dynamic information streams. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 99–106. IEEE, 2009.

[21] L. Shi, F. Wei, S. Liu, L. Tan, X. Lian, and M. X. Zhou. Understanding text corpora with multiple facets. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 99–106. IEEE, 2010.

[22] Y. Song, S. Pan, S. Liu, M. X. Zhou, and W. Qian. Topic and keyword re-ranking for lda-based topic modeling. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1757–1760, New York, NY, USA, 2009. ACM.

[23] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[24] F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 979–988, New York, NY, USA, 2006. ACM.

[25] F. B. Viégas and M. Wattenberg. Timelines - tag clouds and the case for vernacular visualization. *Interactions*, 15(4):49–52, 2008.

[26] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15:1137–1144, November 2009.

[27] C. Wang, D. M. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, UAI '08, pages 579–586, 2008.

[28] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12nd ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, 2006.

[29] M. Wattenberg. Baby names, visualization, and social data analysis. In *Proceedings of the Proceedings of the IEEE Symposium on Information Visualization*, pages 1–7, Washington, DC, USA, 2005. IEEE Computer Society.

[30] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 153–162, New York, NY, USA, 2010. ACM.

[31] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Dirichlet process based evolutionary clustering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 648–657, Washington, DC, USA, 2008. IEEE Computer Society.

[32] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 658–667, Washington, DC, USA, 2008. IEEE Computer Society.

[33] J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *NIPS*, pages 1617–1624. 2005.

[34] J. Zhang, Y. Song, G. Chen, and C. Zhang. On-line evolutionary exponential family mixture. In *Proceedings of the 21st international joint conference on Artificial intelligence*, pages 1610–1615, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[35] J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 1079–1088, New York, NY, USA, 2010. ACM.