Visualizing Evolving City Stories from Streaming News

Category: Application/Design Study

ABSTRACT

We develop a dynamic visualization system to tell city stories extracted from massive city news in multiple years. The news items continuously arriving along time form a text stream, which is rendered by gradually evolving animated visualization aimed to help users observe and understand the dynamical topics, events and trends of urban cities. The dynamic visualization is implemented based on an incremental clustering scheme over an evolving graph consisting of the streaming news in a moving time window. The changing clusters discover thematic evolution of the city over time. They are visualized based on text summarization so that users can easily explore salient and changing focuses of the text stream which narrates the city's chronological progression. We conducted user study to compare the animation approach with typical interactive visualization approach to evaluate our design which achieved positive results. The system can be extended to other text streams to visualize stories from emails, blogs, and so on.

1 INTRODUCTION

Nowadays the majority of the world's population lives in urban cities. The social and economic progression of cities and their citizens is recorded and manifested in such data as news articles, blogs, etc. Visually telling city stories can help people "visually read" the chronological evolution of different communities and cultures. In this paper, we present a dynamic visualization system to visualize such stories extracted from massive news over a long period of time.

Facing the scale and time length of the news datasets, our aim is to allow readers and analysts to observe and understand the changing characterization and thematic trends of the city. Our system is designed for the following tasks:

- Retrieve and present the salient topics and events that represent a city's story;
- Represent contents of the topics and their temporal evolution with effective knowledge representation;
- Provide a story narrating visualization interface for readers.

To fulfill such tasks, we seek help from narrative visualization [26] and text visualization techniques [17]. However, the state-of-the-art techniques are in the preliminary stage to narrate complex and temporally evolving events. Most efforts incorporate storytelling through exploratory interactions with static visual metaphors. On the other hand, animation is an effective instrument when people attempt to capture the phenomenon of dynamics. For example, films illustrate the dynamic stories in people's life and imagination. Animated visualization should be a good way to describe city dynamics as well. In this paper, we use animated visualization to discourse on a city's story so that users can naturally study the chronical evolution of the city in news. The animated visualization follows several design principles:

- *Support theme evolution:* The visualization should handle the streaming news evolving along time and present the changing themes;
- *Discover salient points:* The visual elements should capture critical features and trends from the text stream.
- *Use understandable visualization:* The dynamic visualization need to give users an easy-to-read interface, and avoid complex metaphors and interactions to promote easy understanding.
- *Promote user engagement:* Users should be able to browse the story with minimal effort of learning and exploration.

Our system presents city stories using animated visualization that fulfill these requirements. It is different from major existing methods. Compared to the exploratory tools which ask users to conduct extensive interaction over visualizations, our method narrates a city story by playing gradually evolving animation, in a manner of browsing a book. On the other hand, unlike specific infographics which convey professionally designed information, our approach processes news items and generates visualizations automatically. These features make our method feasible to visualize stories from massive news data, and generally extensible to other text streams.

In this study, we retrieved the news of a set of 30 U.S. cities across more than 10 years from the well-known New York Times news repository. Then our visual system presents dynamic visualizations for users to visually browse the story of their selected city in a given period. The system offers the following features to fulfill the design principles:

- *Dynamic graph model:* News items flow into the system and a dynamic graph is used to represent them and their dynamic relations.
- *Incremental clustering:* To capture salient points of a story, the evolving graph characterizes the news items into varying clusters that discover specific topics to elucidate the evolutionary story.
- *Text summarization:* Instead of displaying lots of individual news items contained in each cluster, text summarization of major clusters is depicted with minimal clutters and distractions.
- *User engagement:* The story can be presented in different levels of details for different cities and periods selected by users. Users can also manipulate the animation with forward/backward/pause functions.

We implemented a prototype for city stories. To evaluate our design of the animated visualization, we performed a user study to compare the performance with the approach using interactive visualization. The study showed positive results about city story animations.

2 RELATED WORK

Stories are usually summarized and presented by timelines, such as in a history summary from Wiki. Popular infographics tools present static views to graphically show the evolution of events. For example, graphical timelines manually extracted from news are widely used by medias. Salient news are also selectively viewed through a table view by Google News Timeline. A treemap view shows the current news in Newsmap.jp, while photos from news are visualized in a similar manner such as by tenbyten.org. However, these tools are not designed for narrating massive news over long period.

Visualizing large text corpora is an emerging research topic [17]. Tag cloud techniques (e.g. [19]) use visual depictions of tags (or words) giving greater prominence to words that appear more frequently. Similarity-based projections help users get insights from large text collections in 2D views. Typically, multidimensional scaling (MDS) or force-directed methods are used to form 2D layout from the text documents. For instance, "galaxies" or "mountains" are formed in the displays [31]. A hierarchy of the documents are built and projected as circles in [25]. InfoSky [3] exploits hierarchically structured documents at each level with Voronoi diagrams. Examplar-based visualization [6] visualizes extremely large text corpus by probabilistic MDS projection with approximation and decomposition.

Text collections often include time-stamped documents. The-

meRiver [13] and LensRiver [12] depict the frequency changes of keywords as river currents. T-scroll [14] employs a novelty-based clustering algorithm on time-series documents. StoryTracker [16] presents a incremental visual analytics system for exploration of news topics in dynamic information streams. Meme-tracking [20] extracts keywords from news corpus to generate themes, and then visualizes them to indicate the flow of stories. Utilizing text mining tools such as topic modeling, some efforts have been made for visual analysis of text collections, including EventRiver [23], Cloudlines [15], Leadline [9], visual text summary [22], TIARA [30], TextFlow [7], Textpool [1], and TextWheel [8]. These methods make great success in visualizing the temporal trends in text archives or streams, mostly through a global view plus interactions. They may not be directly applicable in telling the city story from massive news in a long period.

Using dynamical views to visualize text streams, Alsakran et al. [2] use a dynamic similarity-based projection system to depict text streams. The dynamic evolution of documents inside the 2D projected layout is seamless and smooth. However, this approach visualizes all the particles inside clusters, whose layout is too complex. The dynamic motion of lots of particles is disturbing for users to focus on salient information. Gansner et al. [11] draw the dynamic graph of streaming documents by MDS. They use a map metaphor to depict the clusters, where highly related messages form countries enclosed by a boundary. This approach creates a static layout at a time. To handle dynamic text streams, a two-step approach is used. A newly arriving document is placed at the average of its neighbors in the previous graph and local optimization of MDS layout is applied. However, a special Procrustes transformation has to be applied to make the new graph best aligned to the old one. Moreover, to avoid overlapping components after local MDS, a packing algorithm has to be introduced. The dynamic maps approach [11] including many cities and countries on the screen may also impose heavy burden for observers.

Our approach instead computes and depicts topical clusters and their dynamics of city news through animated strips. It helps users to read interesting topics and their evolution with minimal learning curve.

3 Processing City News Stream

Fig. 1 illustrates the processing framework of our system. A news stream continuously injects arriving documents into the system. Each news document has its title, abstract, and timestamp. It is typically represented by a sequence of keywords, based on which we compute the similarities between pair-wise news. We use the cosine similarity in this computation. For *in situ* computation, we update a current set of keywords once new documents arrive. Not all the keywords are included in the set, while the TF-IDF (Term Frequency-Inverse Document Frequency) weights are used to find the most important N keywords for similarity computation.

3.1 Evolving Graph and Topical Clusters

Based on a force-directed method, a 2D layout of active documents (represented as particles) is automatically generated. Similar documents are placed closely in the layout while dissimilar ones are faraway. Over this 2D layout, the document particles are triangulated which creates an evolving graph. Then we conduct a controllable graph division. The edges having a length smaller than a given threshold are removed. Then the connected documents form topical clusters. This approach, similar to [2], can automatically handle continuously incoming documents and create clusters over these documents in a given time window. It has a mechanism for dropping old and stale documents out of the window.

Figure 2a is the evolving graph created after applying Delaunay triangulation to the 2D layout of news document particles. By applying graph cut with a threshold ϕ, we can discover several clusters

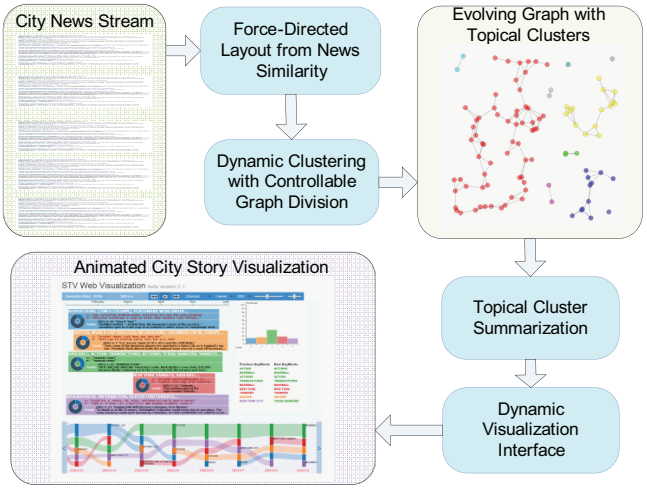


Figure 1: System framework. A news stream continuously injects news items into the system along time, which are processed and visualized for users to characterize and understand the city story.

over the graph in Figure 2b. The clusters can be generated with different settings of ϕ. Figure 2c shows more clusters are generated.

The key problem in mining and visualizing text streams is to identify the evolutionary topics. Such data stream clustering problem requires incremental processing which is reviewed in a survey [27]. Our approach naturally models the topical cluster variation without seeking help from the text mining techniques. It can be completed immediately and controlled by the threshold.

3.2 Evolutionary Topical Clusters

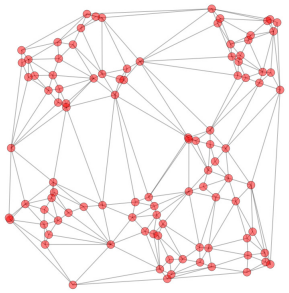
To trace the evolving clusters, it is necessary to match the important pairs of clusters in consecutive time windows, which share most of their news particles. In detail, given two clusters, C_i and C'_j, in consecutive time t and t', each of them accommodates a set of particles. We can compute a match factor δ_{ij} between the two clusters as:

$$\delta_{ij}(C_i, C'_j) = \frac{N(C_i \cap C'_j)}{N(C_i)}, \tag{1}$$

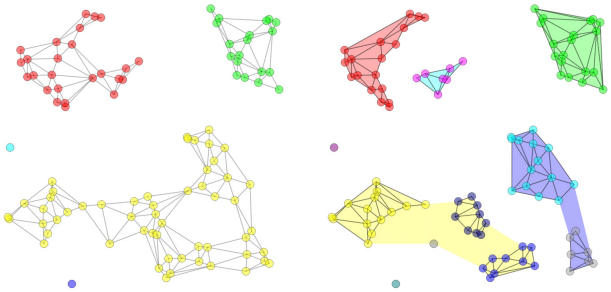
where $N(C)$ is the number of particles inside a cluster C. Then, considering a set of clusters $\{C_i\}, i = 1 \cdots N$ found at time t, and a set of clusters $\{C'_j\}, j = 1 \cdots M$ formed at time t', we need to find the best matching pairs $C_i \mapsto C'_j$ for each C_i. This cluster matching problem can be solved by a combinatorial optimization algorithm, the Hungary method [18].

This method provides a better solution than a naive matching implementation in [2]. In particular, we have a nonnegative $M \times N$ matrix, where the element in the i-th row and j-th column represents $\delta_{ij}(C_i, C'_j)$. It can be looked as the cost of assigning $C_i \mapsto C'_j$. We apply the Hungary algorithm to find an optimal solution where the total cost $\Sigma_i \delta_{ij}(C_i, C'_j)$ of all $i = 1 \cdots N$ is minimized. In our implementation, we further extend the algorithm by setting a lower bound of $\delta_{ij}(C_i, C'_j) = 0.15$ representing the smallest value between any matched pair. Because if the match factor is too small, i.e., the two corresponding clusters do not share enough documents, we cannot match C'_j as the evolved cluster from C_i. Based on this result, we can identify the merging and splitting of the topical clusters from t to t'. Finally, any cluster at time t' that does not have a match from the previous time is considered as a new emerging cluster.

Important clusters represent the thematic knowledge of the news stream, as we can look at each cluster as a topic. In [2], though some clusters are identified, the visualization simply shows individual particles over the force-directed layout. Such visualization



(a) Evolving graph



(b) Topical clusters (c) More clusters

Figure 2: Topical clusters can be achieved immediately from the evolving graph which is easily controlled by a threshold.

is too complex for users to understand the changing graph. Moreover, users cannot easily visualize the information in topical clusters and their relationships. Instead, our visualization design shows the changing news focuses with textual summaries discussed in the next section.

4 Topical Cluster Summarization

Visualizing the topical clusters relies on a fast, automatic way to find and present the most important points of the original news inside a cluster. Due to the limited real estate in visualization, only a very small set of words, phrases, or sentences can be used in the visual representation. For evolving news streams, we utilize three types of information to quickly convey the important content of a topical cluster including:

1. Active time period: Focusing topics in a news stream evolve along time. The active time period of clusters provides information of a cluster's time span.
2. Representative keywords: As a news item is represented by a set of important keywords, a cluster is also represented by some representative keywords. The selection of such keywords can be completed by using the most frequent keywords, or by using the highly weighted keywords (e.g. by IF-TDF).
3. Significant titles: Significant news inside a cluster usually represent salient knowledge. Visualizing their titles helps users understand the topic. Finding the most significant document is implemented in three ways:
   - The most recently (newest) arriving document;
   - The document that has the largest average similarity to other documents;
   - The document has the highest PageRank [24] to other documents;

   Figure 3 is an example of three different news found in a cluster with these three options. It indicates the news particle and the titles of their corresponding news in different colors. The representative hottest keywords are also showed. We implement the algorithms of showing all of these options.

In the future, we will study more summarization techniques (e.g. [32]) to better represent a topical cluster.

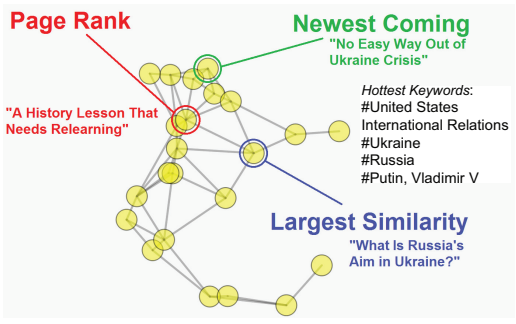


Figure 3: Summarization of a cluster of politics. Hottest keywords are shown. Moreover, three different methods are used to find important documents whose titles are shown in different colors: (Green) the newest coming document; (Blue) the document having the largest average similarity to other documents. (Red) the document having the largest PageRank [24].

5 Visualization Design Rationale

Our goal is to help users visually read a city's story by browsing their news from New York Times. Our visualization design provides an easy-to-read dynamic view of the important topics discovered by the evolutionary clusters. Our design rationale in a context of existing methods is explained in the following:

- **C1:** First, this task cannot be accomplished by showing all news to users with their raw contents including title, abstract, and first paragraph. This approach needs excessive effort from users to read a large set of text given a long period (e.g. one year with thousands of news). Meanwhile, this task also cannot be achieved by extracting frequent keywords which does not provide enough story.
- **C2:** Second, existing news visualizations, such as using a table view with timeline (e.g., Google News Timeline) or a treemap view (e.g, Newsmap.jp, and 10X10 tenbyten.org) cannot perform well in story telling. These tools allow users to interact with the static views to read salient, individual news in the present time or in a short nearby period. They do not easily apply to lots of news in one or multiple years, since users will have to extensively find and read interesting news items.
- **C3:** Third, many recent text visualization techniques, such as [7, 23, 9, 8, 28] and more [21], adopt a strategy that uses a global view (e.g., themeriver, textwheel) to provide high-level abstraction of news topic/keyword evolution, and then ask users to interact with this view to discover details. Before visualization, they conduct semantic clustering, event detection, LDA, and more methods in text mining (or machine learning) for the information aggregation and abstraction from massive documents. Then these methods require users to arbitrarily manipulate the visualizations for interactive knowledge discovery. We appreciate these techniques which have achieved success in many text stream visualization cases. In our design, we want to shorten the learning curve of users to study the specifically-designed visualizations, and reduce their interaction efforts. These issues may easily become a hurdle to general public when browsing a city's story. Therefore, we try to use animated view of dynamically computed clusters to allow users directly reading the topic contents, without much prior investment on understanding and learning the visualization and interaction.
- **C4:** Finally, some existing work seeks animated view to visualize

text stream, such as [2, 11]. The dynamics can give users a direct visual cue on understanding temporal changes in a story. However, in visualizing dynamic scenes, too many visual elements on the interface usually confound observers' perception and understanding, since the changes of complex visual metaphors can easily become disturbing, especially for textual information. Many dynamically moving particles [2] or varying spatial regions [11] can distract users' focus and reading efforts. A balance between static visualization and dynamic views is applied in our animation method.

The four approaches **C1-C4** have their pros and cons for city story telling. The methods in C1 and C2 do not consider topic discovery among massive news so they are not qualified for city stories. Most methods in C3 forms the mainstream strategy in recent text visualization techniques. This strategy projects the spatial-temporal news data into spatial visualizations that include explicit time coordinates (such as the time axises in theme river), and then allows users to interactively explore the visualizations for detail knowledge discovery. Next, we refer to this strategy as *C3-Strategy*. The animated methods in C4 are not popular since the effectivity of animation in text stream visualization needs more justification in comparison to the C3-Strategy. In this paper, we design the new method to test whether and how the animated visualization can help in the city story scenario, which can also be extended to other text stream visualization cases. Our major goal of design is to provide an animated visualization interface that is:

- *lightweight:* with a limited number of visual elements that do not overwhelm and defocus observers during evolution;
- *informative:* including necessary information for users to read and identify the topical contents;
- *smooth:* with minimized abrupt changes or excessive user interaction during dynamic topic evolution.

Next we show our visualization interface to fulfill this goal. Then we show a case study for a city story. We further present a user study to comparably investigate the difference and performance between our approach and a topic river view that follows the C3-Strategy (Sec. 9).

6 City Story Animated Visualization

6.1 Dynamic Animation

Our visualization interface is fed by the dynamically generated topic clusters discussed in Sec. 3. In implementation, users first choose their interesting city to be read. They also set up the total time period (P) which they want the city story to be shown, such as one year or multiple years. The story is narrated in a series of frames, each of which shows the contents of a specific date. A current visualization frame of date T^A shows the topic clusters generated from an active pool of news belonging to an active period (t_a) before T^A. After a fixed reading time (e.g., R), the next visualization frame updates the view which shows the changed topics, which are created by updating the pool with a set of news happened in a time interval (t_i) after T^A. Now the current date advances to T^B=$T^A + t_i$. Here, old news is removed from the active pool to make sure that all news items in the pool belong to t_a before T^B. For example, if setting t_a = *two months* and t_i = *one month*, a frame of T^A = 07-31 (We use the mm-dd date format in this paper) shows the news topics for both June and July. Next animation step, a new frame of T^B = 08-31 shows the topics for both July and August, while news in June is removed from the active pool. Between these frames, we provide a little transition effect to make the change smooth and beautiful which however is not too disturbing for readers.

t_a and t_i define the time window and the resolution of story telling. A weekly setting with t_i = *one week* creates 52 visualization frames in the animated visualization for a period of one year. Using t_i = *one month* interval instead leads to 12 frames for the one year's story. Users can adjust these parameters to browse the story in different ways. Between frames, users can also choose the length of reading R. If they want to follow an important topic and its evolution quickly, R can be short. When they want to study more topics, R can be set longer to have more time on reading a frame.

In summary, our dynamic visualization automatically provides an animated view mimicking the slide showing effect. It allows users to read the content, together with the time evolution.

6.2 Visualization Interface

Figure 4 displays the city story visualization interface on one frame. Users first select the city ("Austin") and time period P (2003) in the control panel (Figure 4(D)). When the animation starts, users can play forward, backward, and pause the process by the panel.

One visualization frame is illustrated in Figure 4(A). Five topical clusters are visualized in distinct colors, each as a strip. A time ruler above them shows the time range of each topic. The strips are ordered from top to bottom according to the size (i.e., the amount of news items) of the clusters. The top one is the most salient topic who has the largest number of news in the active period t_a before the current date. T is dynamically shown in the panel of Figure 4(D) as time evolving. Each strip further shows the summarized information of the cluster which is discussed in Sec. 4. At the left side of a strip, in the middle of a pie icon, the number of news items in the corresponding cluster is shown. this icon displays the percentage of this amount among all these clusters. The percentage distribution of these clusters are further shown in Figure 4(B) to help users quickly see the topics' importance.

Along the animation between frames, a strip changes its length measured by the time ruler. A cluster may change its vertical position due to the change of its size. When a topic is no longer significant, that is, no longer the top 5 clusters, it will be removed from the view. A new cluster replacing an old one will be shown as a new strip which is highlighted to reflect its emergence. We fix the colors of these clusters. A new cluster will replace the removed one with the same color (but highlighted). This approach has less distraction than giving each new cluster a new color. In the view we keep the top 5 clusters/strips. This number can be adjusted by users as well.

Figure 5 is one red topic extracted from Figure 4(A). This topic cluster has a range roughly from Febuary to June, which is the current date being read during the animation. On the top of this strip, the representative keywords are shown to represent the major themes of this cluster. This topic is about entertainment and culture including music, movie, dancing, etc. It currently has 21 news. In the middle, the title of the largest similarity news (S) and the title of the highest Pagerank news (P) in this cluster are shown starting with S and P, respectively. Users can identify such representative news in this topic. Moreover, the latest arriving news is displayed in more details, showing its date, title, and first paragraph. This example shows a news about an event in the Museum of Contemporary Art. Note users can choose to replace this detailed news with S or P news according to their interest.

Figure 4(C) displays a list of top keywords among all active news, after inserting the news items in a new frame. This list is led by "New". In comparison, the previous list of keywords is shown by "Previous". In this visualization, a keyword is colored by the same hue as the cluster it belongs to in the corresponding strip. Keywords appear only in "New" is highlighted by adding a boundary so that readers can identify emerging words. For example, a term "ATOMIC WEAPONS" becomes visible in the purple cluster, which refers to the strip of US Defense and Iraq. In this way, the two lists are compared to help users find the difference introduced by newly inserted news.

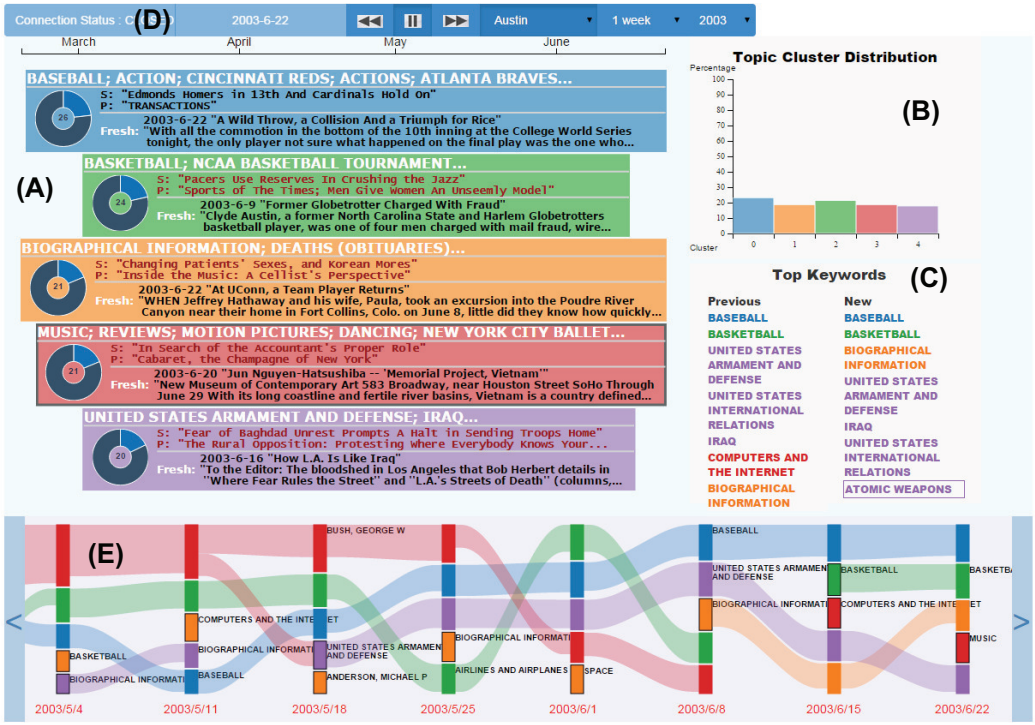


Figure 4: City story visualization interface. (A) Dynamic view of topic clusters; (B) Cluster distribution view; (C) Top keywords view; (D) Control Panel; (E) Topic river view.

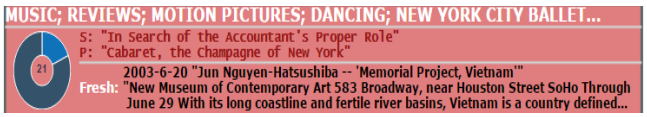


Figure 5: Visualize summarized information of one topical cluster.

6.3 Topic River View

We further design a topic evolution view to provide a context to the animation view. This view is based on the theme river, to show the evolution of topic clusters. As shown in Figure 4(E), the topical clusters are ordered from top to bottom by the size at each time axis, and then connects similar topics by ribbons. The width of ribbons relates to the size of clusters. Major keywords in the topics are shown over the ribbons. The ribbons stream forward automatically along time during dynamic animation. Users can directly observe a topic's change, including emerging, splitting, merging, and disappearing. Moreover, this view can be dragged backward and forward. Such visualization helps users better observe the thematic changes in a story. In this way, we incorporate the benefit of global view (following the C3-Strategy) into our prototype.

7 System Implementation

To maximize the usability, we develop the system in a realtime client-and-server scheme. In particular, after a news stream injects new documents into the system server, the documents are processed for cleaning and keywords extraction. Then the server-side program updates the active list of the important keywords, performs *in situ* similarity computation, and uses the force-directed algorithm to create the 2D layout of particles. Dynamic clustering are then performed (Section 3), followed by the summarization (Section 4). Finally, a set of clusters are created while each cluster stores the summarization information. All these operations on the server are implemented by optimized C++ programs for fast performance.

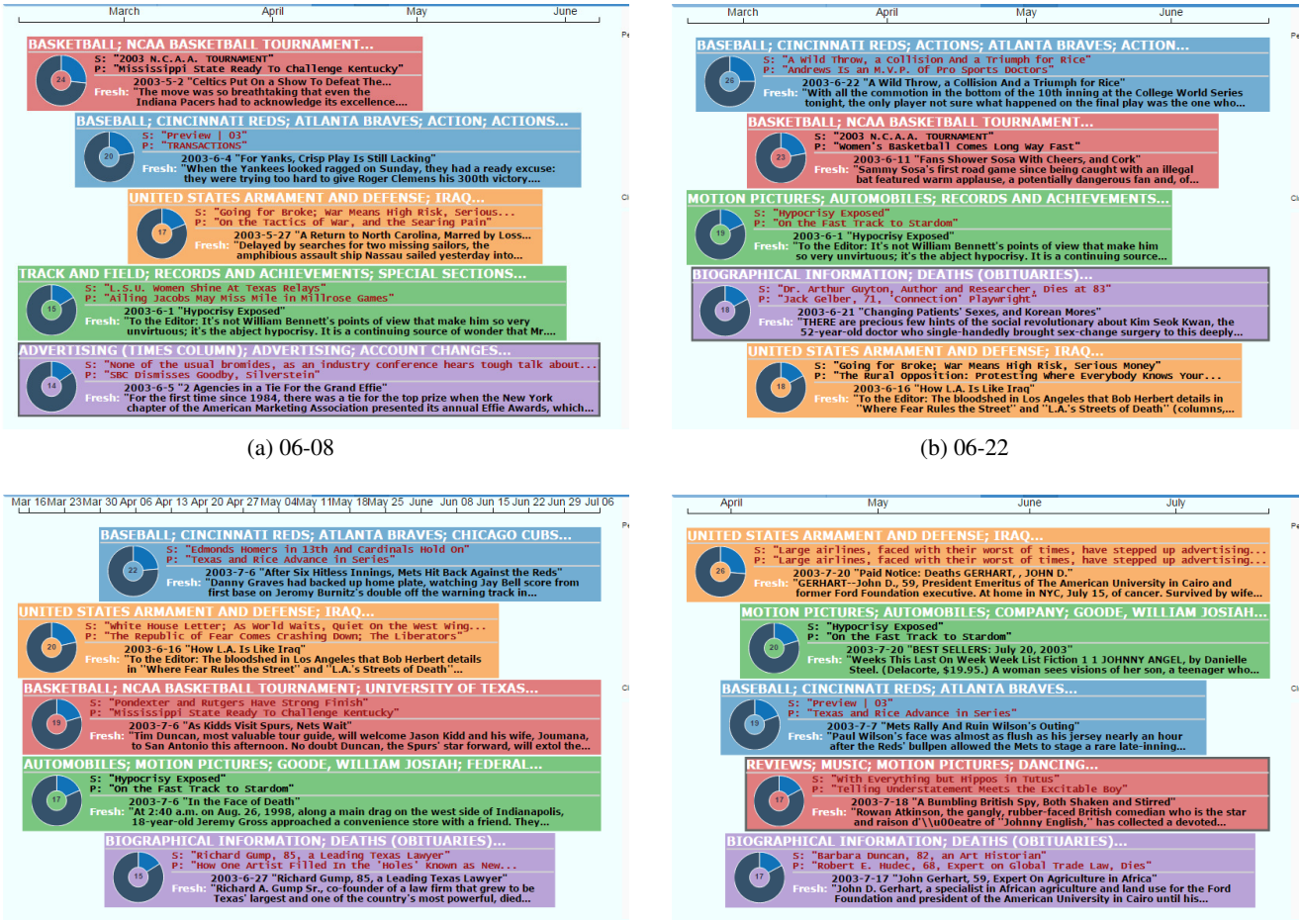
The server starts and maintains a realtime data exchange service. One or more web clients can be initiated by linking them to the server with internet connection. The data exchange service between the server and clients is implemented through the JSON (JavaScript Object Notation) format [10], a text format that is completely language independent. The communication between server and client is completed by Web Sockets API [29] that enables web pages to use the Web Socket protocol for full-duplex communication with the remote server. The full-duplex channel is important since in realtime we transfer control parameters adjusted by users to the server side to generate different results according to their input.

A client's task is to provide web-based visualization. It receives the information of the multi-clusters continuously and then visualize them on web pages. The visualization is completely web-based, implemented by Javascript and HTML5 with D3js [5].

8 Case Study: Story of Austin Texas

In this case study, we explore a news stream collected from New York Times (www.nytimes.com) about the city of Austin, Texas. The news stream includes 795 news mentioned Austin in the year of 2003. This stream reflects Austin in the eye of New York Times which may not include all news interesting for residents.

Each individual news document includes the keywords, published time, title, and the first paragraph of the news. We mimic continuously arriving news items flowing into our system by their publishing time. We first set the parameters (Sec. 6.1) as *P=one year*, $R = 15$ *seconds*, $t_a = 12$ *weeks*, $t_i =$ *two weeks*. Figure 6 shows four frames of the biweekly Austin story of the specific dates between June and July. This figure only displays the strips of clusters for clarity. From Figure 6a of the date 06-08, two top topics are found which are about basketball and baseball, respectively. Basketball has more related new (24) than baseball (20). The latest Basketball news is at May (05-02). This cluster is mostly about NCAA basketball which is more interesting to the city. In comparison, a baseball news comes at 06-04 which is more recent. Two weeks later, the baseball topic becomes more significant with more



(a) 06-08

(b) 06-22

(c) 07-06

(d) 07-20

Figure 6: Case study: City story snapshots from a biweekly animation of Austin, Texas in 2003. Each figure shows a frame of the specific date.

news items than basketball. As shown in Figure 6b, the two strips swap their position. Users can also find some detail about women's basketball from the P title. In Figure 6c, baseball keeps its hotness while basketball decreases. The fresh news at 07-06 talks the sport detail meaningful to baseball fans. Meanwhile, an emerging topic about US defense and Iraq moves up. After another two weeks, by 07-20 basketball is no longer a major topic for the city while the baseball topic falls to the third place. The latest news about baseball is at 07-07 which explains this fall. In the supplemental video, this animated visualization is clearly illustrated in the animation. Please note that we record the video in a fast speed to reduce its size for this paper.

Figure 7 shows a different weekly animation with t_i = *one week*. It shows the frame at the same date 06-08 as Figure 6a. From this figure, the baseball topic is on top of the basketball topic, which is different from Figure 6a. This is because the baseball news items are relatively closer to 06-08 than that of the basketball news. A third-place topic shows Biographical Information, which people can find Obituaries in the P and S and fresh titles. This information does not show in Figure 6. in In this way, with the reduced resolution from biweekly to weekly, the story shows a higher level of details.

9 User Study

We conducted a user study to evaluate our newly designed animation visualization in comparison to the topic river approach following the C3-Strategy. The major goal was to test the performance

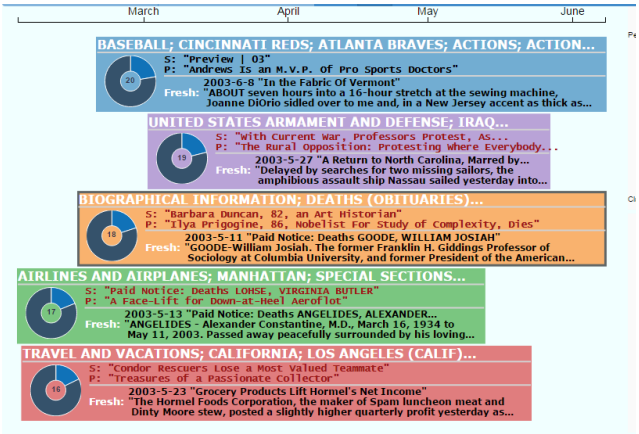


Figure 7: One frame of the Austin city story from weekly animation, at the same date 06-08 as Figure 6a.

of the two approaches to justify the use of the animation in story telling. The study used the monthly evolving clusters created from the Austin dataset in 2003 by defining t_i = *one month* .

9.1 Two test settings

On two different machines, we used two visualization approaches:

- *Animation*: We showed the animation as in Figure 4(A). Here the view of Figure 4(E) was not shown. Then this setting was a pure animation visualization with the topic strips.
- *Topic river*: We showed only the river view like Figure 4(E) for the whole year with 12 axises. Users could interactively move the mouse over interesting topics to see the detail keywords and major titles. This view then mimicked a simple but typical text visualization approach in the C3-Strategy, which uses visual interaction to help users discover stories.

Please note that our prototype includes both of these views, but in this study, we aimed to justify the use of animation compared with interactive visualization by testing users' performance over these two settings, independently.

9.2 Participants

The study was conducted with 18 participants (4 females and 14 males). They were undergraduate and graduate students from different departments in a university, while a majority of them was from computer science. Participants were between 21 and 33 years old. Most of them had a basic understanding of data science but not on information visualization. They had not used any text stream visualization system before. Two instructors worked on this project instructed them to perform the study.

9.3 Tasks and Process

We first randomly divided these participants into two groups (each had 2 females and 7 males). One group was assigned to the animation setting, and another group to the topic river setting. Participants were required to finish the tasks in two stages:

- Stage 1: They read a short written description of their corresponding visualizations. The short description simply explained the meaning and the order of the strips for the animation setting. For topic river setting, it described the meaning of the time axis, cluster bar and their order.
  Then Group 1 browsed the animation, and Group 2 interactively visualized the topic river. They answered two questions of Q1: *(a) Write down the top 2 important topics for the date T^A and T^B; and (b) write down the emerging time of a given cluster, E_1, and the time it becomes the hottest topic.*
- Stage 2: Instructors gave an interactive training of the visualization and interaction, until the participants felt confident on fluently using them. Then they performed the similar process to answer the two questions of Q2: *(a) Write down the top 2 important topics for the date T^C and T^D; and (b) write down the emerging time of a given cluster, E_2, and the time it becomes the hottest topic*. These two questions are the same to Q1 in Stage 1. In Q2, they also were asked the third question to *(c) Write down E_2's keyword details.*

Stage 1 was designed to test the performance of the two settings when no detail training is given. It reflects the ease of use for the visualizations without a long learning time. In contrast, Stage 2 tested the performance of the two settings when enough training was given. For measurement, we recorded the times (in minutes), T1 and T2, which were used by each participant to complete Q1 and Q2, respectively. They were also given a score of Q1 (0-20) for its two questions and Q2 (0-30) for its three questions. The scores were given by comparing their answers with ground truth.

Ideally, we hoped a participant could reach higher scores and use shorter time in their test. These two metrics can be combined into one measure – the score-over-time computed as Q1/T1 and Q2/T2, which reflected the score achieved per minute by a participant.

After finishing the two test stages, we further collected feedback from the participants about two questions: (F1) *whether the topical clusters can be identified clearly*? and (F2) *whether the evolution of topics can be better discovered*? They answered the question with a rate from 0-2, where 2 was yes and 0 referred to no.

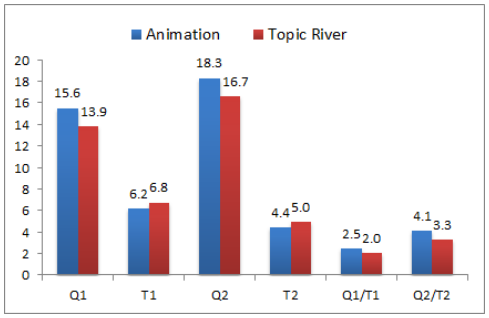


Figure 8: Evaluation results for the two settings, including the average scores of Q1 and Q2, the average time used of T1 and T2, and the score-per-minute of Q1/T1 and Q2/T2.

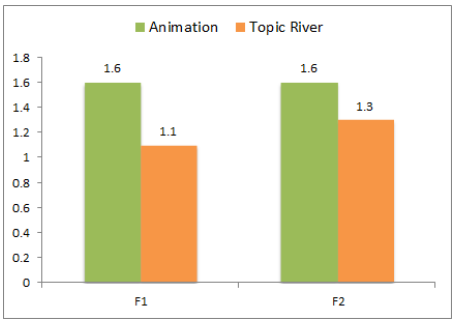


Figure 9: Feedback ratings from the participants about two visualization settings.

9.4 Results

Figure 8 shows the evaluation results in Stage 1 and 2 for the comparison of animation and topic river. In Stage 1, the animation group achieved a better average score (15.6) than the topic river group. The animation group also used less average time (6.2 mins) than the topic river group (6.8 mins). In Stage 2, the animation group also performed better than the topic river group with a higher average score and a smaller average time. The animation group also had a larger score-per-minute (Q1/T1) than that of the topic river group. Therefore, in the given event discovery tasks, the animated strips showed better performance in a row than the topic river group. However, the performance differences in both stages between the groups are less than 15%. Another observation was that the score was higher and the used time was shorter in Stage 2 than in Stage 1. This agreed with the estimation that the training session improved the visualization performance.

Figure 9 shows the feedback from the participants after they finished the test. For the question F1, the average rating they gave to the animation is 1.6, while it was 1.1 for the topic river. For F2, the animation visualization got 1.6 and the topic river reached 1.3. These values showed that the animation was scored 45% (F1) and 23% (F2) better than the topic river. These ratings showed that the animation achieved better evaluation from the participants. Note that they were not given any hints on preference to the two visualizations from instructors.

9.5 Statistical Test

We further conducted a two-tail t-test to justify the statistical significance of using the two visualization settings. We used the score-per-minute (both Q1/T1 and Q2/T2) values of the two groups to perform the test, because this measurement combined the two metrics of score and time. We got p-value = 0.0124. Therefore, we

rejected the null hypothesis that the means of the two settings are equal. We concluded that the means of score-per-minute when using the two settings differed significantly.

9.6 Summary

The user study primarily showed that our animated visualization played well in story telling tasks from the city news stream. In the comments of the participants they pointed out what can be improved by more delicate design in visual representations, such as color, label and animation control. These improvements will be conducted in our future work.

10 Conclusion

We have presented a visualization prototype to visualize city stories from news streams. It facilitates real-time topic discovery over streaming data, uses easy-to-understand visual metaphors, and promotes user understanding of the thematic topics and their evolutions. We conducted a user study which primarily showed that the new animation visualization can help users in conducting knowledge discovery tasks. The user study was performed on the animation in comparison to an interactie visualization method. Moreover, the feedback from users gave positive ratings to the animated story telling technique. Motivated by these results, we believe that our work will have a positive impact on the visualization of evolving data stream. The approach can be used in visual exploration of news stories, blogs, emails, business transactions, and more text datasets. In the future, we will study more topic and cluster summarization techniques. We will also perform more extensive user studies over animated visualization techniques.

References

[1] C. Albrecht-Buehler, B. Watson, and D. A. Shamma. Visualizing live text streams using motion and temporal pooling. *IEEE Computer Graphics and Applications*, 25(3):52–59, 2005.

[2] J. Alsakran, Y. Chen, D. Luo, Y. Zhao, J. Yang, W. Dou, and S. Liu. Real-time visualization of streaming text with a force-based dynamic system. *IEEE Comput. Graph. Appl.*, 32(1):34–45, Jan. 2012.

[3] K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer, and K. Tochtermann. The infosky visual explorer: Exploiting hierarchical structure and document similarities. *Information Visualization*, 1(3):166–181, Dec. 2002.

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

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[6] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus (infovis2009-1115). *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1161–1168, 2009.

[7] 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 Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, Dec. 2011.

[8] W. Cui, H. Qu, H. Zhou, W. Zhang, and S. Skiena. Watch the story unfold with textwheel: Visualization of large-scale news streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):20, 2012.

[9] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 93–102, 2012.

[10] Ecma-International. Ecma-404 the JSON data interchange standard. http://www.json.org, 2013.

[11] E. R. Gansner, Y. Hu, and S. C. North. Interactive visualization of streaming text data with dynamic maps. *Journal of Graph Algorithms and Applications*, 17(4):515–540, 2013.

[12] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky. Newslab: Exploratory broadcast news video analysis. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 123–130, 2007.

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

[14] Y. Ishikawa and M. Hasegawa. T-scroll: Visualizing trends in a time-series of documents for interactive user exploration. *Lecture Notes in Computer Science*, 4675:235–246, Nov. 2007.

[15] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: compact display of event episodes in multiple time-series. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2432–2439, 2011.

[16] M. Krstajic, M. Najm-Araghi, F. Mansmann, and D. A. Keim. Incremental visual text analytics of news story development. In *IS&T/SPIE Electronic Imaging*, pages 829407–829407. International Society for Optics and Photonics, 2012.

[17] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. *IEEE Pacific Visualization Symposium*, pages 117 – 121, 2015.

[18] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[19] B. Lee, N. H. Riche, A. K. Karlson, and S. Carpendale. SparkClouds: Visualizing trends in tag clouds. *IEEE Trans. Visualization and Computer Graphics*, 16(6):1182–1189, 2010.

[20] 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*, pages 497–506, 2009.

[21] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: Recent advances and challenges. *Vis. Comput.*, 30(12):1373–1393, Dec. 2014.

[22] 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*, pages 543–552, 2009.

[23] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. Keim. Eventriver: Visually exploring text collections with temporal references. *Visualization and Computer Graphics, IEEE Transactions on*, 18(1):93–105, 2012.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical Report*, 1998.

[25] F. Paulovich and R. Minghim. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transaction on Visualization and Computer Graphics*, 16(8):1229–1236, Nov. 2008.

[26] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1139–1148, 2010.

[27] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1):13:1–13:31, July 2013.

[28] C. Wang, Z. Miao, S. Chen, Z. Liu, Z. Wang, Z. Wang, and X. Yuan. Story explorer: A visual analysis tool for heterogeneous text data. *IEEE VAST Symposium, Poster*, pages 335–336, 2014.

[29] V. Wang, F. Salim, and P. Moskovits. *The Definitive Guide to HTML5 WebSocket, Build Real-Time Applications with HTML5*. Apress, 2013.

[30] F. Wei, S. Liu, Y. Song, S. Pan, M. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proc. KDD*, pages 153–162, 2010.

[31] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information for text documents. *Readings in information visualization: using vision to think*, pages 442–450, 1999.

[32] X. Zhu, A. Goldberg, J. V. Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. *HLT-NAACL*, pages 97–104, 2007.