RankExplorer: Visualization of Ranking Changes in Large Time Series Data

Conglei Shi, Weiwei Cui, Shixia Liu, *Member, IEEE*, Panpan Xu, Wei Chen, and Huamin Qu, *Member, IEEE*

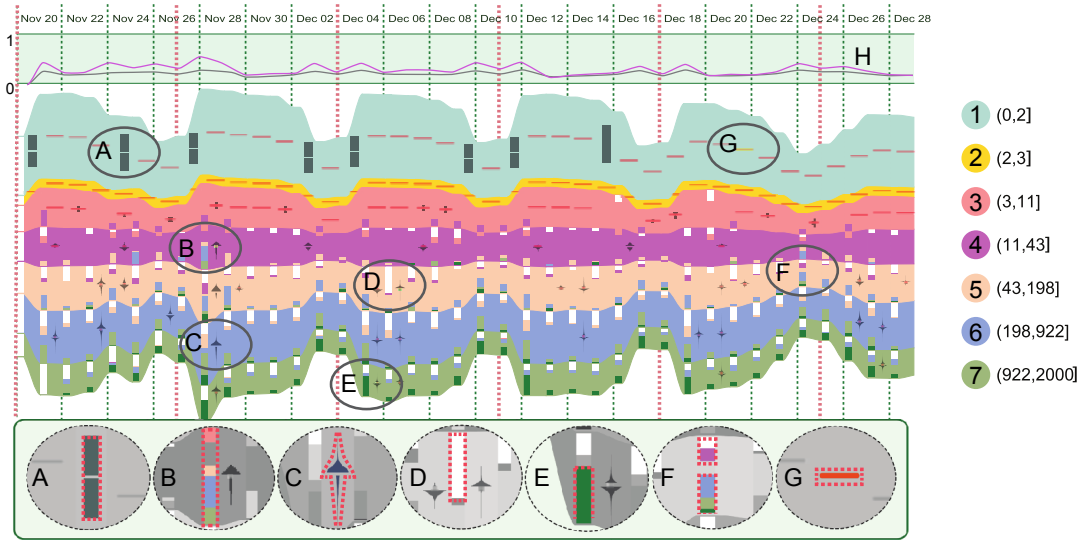

Fig. 1. RankExplorer visualization of the top 2000 Bing search queries from Nov. 20 to Dec. 29 in 2011. All queries are divided into seven categories. The width of each layer at a time point encodes the total query count at that time. The color bar and glyphs encode the content changes in each ranking category. From the color bar, we can observe: 1) the change between layers (the bar segments with the colors of other layers in B and F); 2) new queries coming in (the white segment in D); 3) recurring queries (the dark green segment in E). From the changing glyphs, we can see: 1) a non-change pattern (only red line in G); 2) a swap pattern (the two equal-height segments in A represent that the two queries swap their rankings); 3) a shift pattern (the increasing part is significantly larger than the decreasing part in C). From the trend curve (H), we can see the degree of ranking change over time.

Abstract—For many applications involving time series data, people are often interested in the changes of item values over time as well as their ranking changes. For example, people search many words via search engines like Google and Bing every day. Analysts are interested in both the absolute searching number for each word as well as their relative rankings. Both sets of statistics may change over time. For very large time series data with thousands of items, how to visually present ranking changes is an interesting challenge. In this paper, we propose RankExplorer, a novel visualization method based on ThemeRiver to reveal the ranking changes. Our method consists of four major components: 1) a segmentation method which partitions a large set of time series curves into a manageable number of ranking categories; 2) an extended ThemeRiver view with embedded color bars and changing glyphs to show the evolution of aggregation values related to each ranking category over time as well as the content changes in each ranking category; 3) a trend curve to show the degree of ranking changes over time; 4) rich user interactions to support interactive exploration of ranking changes. We have applied our method to some real time series data and the case studies demonstrate that our method can reveal the underlying patterns related to ranking changes which might otherwise be obscured in traditional visualizations.

Index Terms—Time-series data, ranking change, Themeriver, interaction techniques.

1 Introduction

Time series data analysis plays an important role in many applications such as finance and business. Understanding trends, motifs, relationships, and anomalies in large time series data provides essential knowledge for many data analysis tasks, including performance analysis, prediction, fraud detection, and decision support. One of the important tasks is to study ranking change patterns among multiple time series. This type of analysis plays an important role in interpreting data, examining the major cause(s) of an unexpected event, and forecasting future circumstances (e.g., future stock movement). For example, top queries in public search engines, such as Google and Bing, are usually adopted to represent hot topics searched on the Internet. To examine the major reasons behind traffic fluctuations, search engine analysts often study the frequency changes of those top queries over time, including the absolute numbers and their rankings. As a result, there is an increasing need for a visual analytics solution to analyze ranking changes in large amounts of time series data.

How to visually present temporal changes over time is a challenge of considerable interest, for which a lot of research [1] has been done.

- *C. Shi, P. Xu, and H. Qu are with the Hong Kong University of Science and Technology. E-mail: {clshi,pxu,huamin}@cse.ust.hk*
- *W. Cui and S. Liu are with Microsoft Research Asia. S. Liu is the correspondence author of this paper. E-mail: {weiwei.cui, shixia.liu}@microsoft.com*
- *W. Chen is with Zhejiang Univeristy. E-mail: chenwei@cad.zju.edu.cn*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Among them, the stacked graph [5, 9, 21, 22] is a widely used technique for its compactness and pretty good summarization of the overall and individual temporal trends for time series data. However, traditional stacked graphs cannot intuitively convey the ranking changes between the data items over time, since the order of each layer is fixed once it appears in the visualization. Even though we can enhance stacked graphs by changing the vertical orders of layers at the time points when the rankings do change, such improvement is not effective due to the following reasons: For a very large time series dataset, visual clutter will become a big problem when the rankings of these data items often change at different time points. Although we can leverage some clutter reduction techniques like clustering to deal with data of high volume [6], they still may fail to properly organize the temporal data with frequent ranking changes over time. For example, the search frequencies of the top 100 queries may change over time, while the list of the top 100 queries may also change. A static clustering technique without considering the changes may easily overlook the correlations between such a pair of changes, and the patterns hidden inside. Even when traditional clustering methods successfully categorize the top search queries into several categories at each time point, they hardly support the analysis on the evolution of the ranking changes inside each category (inner change), let alone the content changes (e.g., series flowing in or flowing out) across different ranking categories (outer change).

To tackle these challenges, we have developed RankExplorer, a new visualization method based on ThemeRiver [9] to reveal the ranking changes within and across multiple categories. In this work, we aim to preserve the intuitiveness and familiarity of stacked graphs while addressing their shortcomings in conveying the evolution patterns of ranking changes inside a large time series dataset. To achieve this, we first segment data items into a manageable number of ranking categories. Essentially, it meets the following two criteria: 1) minimizing outer changes between different ranking categories; 2) averaging the height of each layer. Moreover, we also allow users to flexibly choose segmentation criteria according to their task requirements. Then we enhance the traditional ThemeRiver visualization with embedded color bars and changing glyphs to show both the outer and inner content changes. In order to provide a high level summary of ranking changes over time, we also design a trend curve to reveal the overall change degree of the selected ranking categories. Finally, rich user interactions are provided to support coherent exploration of ranking changes.

The major technical contributions of this work are as follows:

- We extend ThemeRiver visualization with embedded color bars and changing glyphs to convey both the outer and inner changes over time.
- We propose an adaptive segmentation method, which sequentially divides the time series data into a specified number of groups for efficient understanding.
- We provide rich interactions to allow users exploring ranking changes at different levels of detail and from different aspects (i.e., changes within a ranking category and across categories).

2 Related Work

Much effort has been devoted to the visual analysis of time series data. Previous work was systematically surveyed in [1, 4, 24, 29]. Compared with them, our work is directly related to the research on interactive, time-based visual exploration, which can be roughly divided into two categories based on their highlights.

2.1 Visual Representation

The most popular method to visualize time series data is to use line charts and their variants. The line chart was first introduced by Playfair [26] in 1786. Recently, various techniques, including Horizon graph [11], SparkClouds [20], and Braided graphs [14], have been proposed to improve its usability and expressiveness. Javed et al. [14] also compared simple line charts with other three variants to find the best application scenarios for each representation. However, they do not handle scalability well. When the data contain thousands of time series, the visualizations all become very cluttered and hard to interpret.

Stacked graphs [2, 5, 9, 34], as a variant of the traditional line chart, are also very popular for visualizing time series data. In a stacked graph, multiple series are represented as layers stacked one on another. The variation in the width of each layer represents the value changes of each series. Such visualization can provide a clear overview for users to track the trend of each individual series, as well as all the series together, over time. To enhance stacked graphs, several extensions have been proposed to show information beyond the overall and individual trend changes. For example, Shi et al. [27], filled the empty space inside each layer with word clouds to visually summarize a large text corpus. Cui et al. [3] improved stacked graphs by adding the splitting/merging branches between layers to show users the inter-layer relations during their evolutions.

Other types of related work include pixel-based methods [7, 15, 16, 19, 23, 35, 38]. They put colors on various visual elements, such as lines [15], bar charts [7], and matrices [8], to encode additional information in the backend time series data. For example, Keim et al. [15] used a recursive scheme to arrange pixels for illustrating a large group of time-varying data. Hao et al. [8] filled the cells of a multiresolution matrix with different colors to represent the magnitude of the values behind each cell.

RankExplorer originates from the stacked graph. We augment it with pixel-based techniques to intuitively convey the relations (e.g., series flowing in or flowing out) across different ranking categories. With the pixel-based techniques, we reduce visual clutter caused by the series flowing in or flowing out. Ranking glyphs are also designed to illustrate the inner ranking change inside each category.

2.2 Exploration Techniques

Another category of research focuses on introducing exploration techniques to help users quickly identify patterns and analyze content in time series data. As a common adopted paradigm, interactive clustering or aggregating can reduce visual clutter and help users understand data at different granularities. This can be done either in coordinated views [22, 23, 32] or single views [30, 31]. For example, van Wijk and van Selow [30] aggregated time series data in a calendar style to reveal patterns and trends on multiple time scales (daily, weekly, or monthly) through similarity clustering. LiveRAC [23] showed multiple views of the time series data in a grid-based layout so that users can easily compare different series side-by-side at multiple levels of details. Recently, several lenses [7, 13, 18, 36, 37] have been proposed for visual analysis of time series data. They aim to provide users with rich interactions, such as selecting, filtering, zooming, transforming, and aggregating, for data exploration. For example, Hochheiser et al. [12] introduced a widget called Timebox to help users interactively query time series data. The most recent work is ChronoLenses, proposed by Zhao et al. [37], which provides a set of interactions to support exploratory tasks, such as derive new time-series transformation result from the original data.

3 System Overview

The aforementioned techniques mainly focus on exploring the individual time series. While RankExplorer aims to provide several novel interactions specifically designed for examining the interrelations between different ranking categories, as well as the inner changes in each category.

Another related work is visualizing incomplete and partially ranked data, proposed by Kidwell et al. [17]. However, their work cannot handle time attributes in the datasets. Thus it is not suitable for our application.

Following the information seeking mantra "Overview first, zoom and filter, then details on demand" [28], we designed RankExplorer to reveal both inner and outer ranking changes in time series data. Fig. 2 shows an overview of our RankExplorer system. First, our system computes the ranking of each item at each time point and also statistics about overall changes. Then, at each time point, the data is segmented

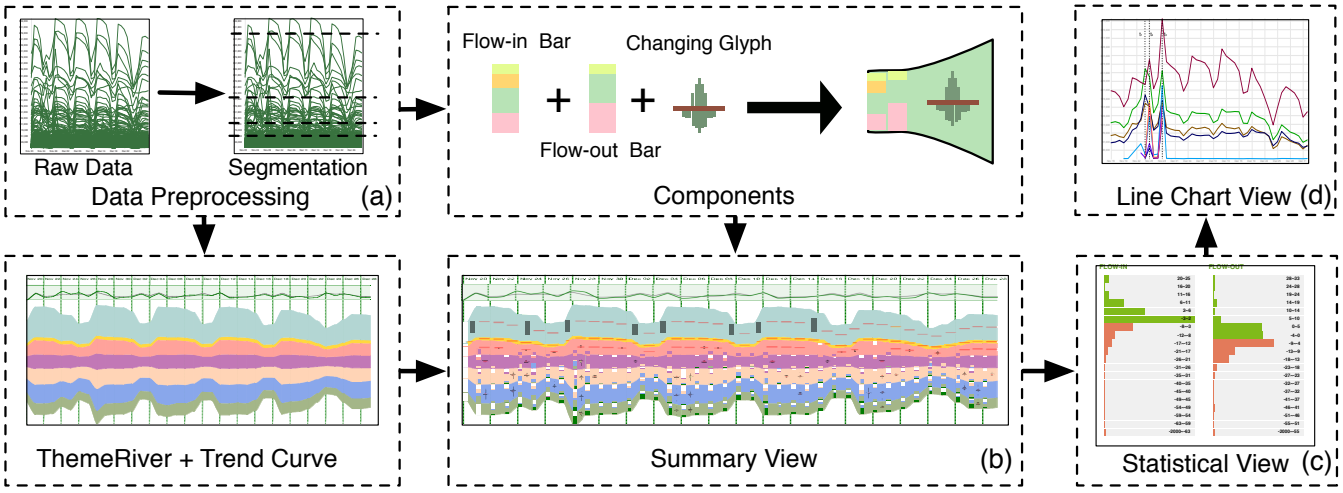

Fig. 2. System overview: (a) data preprocessing; (b) summary view; (c) statistical view; (d) line chart view.

into several ranking categories (e.g., top 1-5, 5-15), while an aggregation value (e.g., total occurrences of the queries) for each category is computed (Fig. 2(a)). The content in each category may change over time (e.g., an item is ranked No. 1 at one time point but slightly drops to No. 3 at the next time point). On the other hand, some items with dramatic changes may even move into or out of the category from one time point to the next, which we call flowing-in and flowing-out items, respectively. For them, our system will also compute statistics related to the content change across categories.

The ranking categories and related statistics are displayed in the *summary view* (Fig. 2(b)), which contains an extended ThemeRiver visualization and a trend curve on the top. The *summary view* provides an overview of the quantitative changes and content shift caused by the ranking changes inside each category and across categories. Rich interactions such as selecting, zooming, and filtering are provided to allow interactive exploration of the data. For example, a user can zoom into a category layer, so that the layer will be expanded to provide more detailed information. For each time point of the selected category, we also provide a *statistical view* to show some key statistical information such as the items number in this category whose rankings go up (Fig. 2(c)). Furthermore, if users are interested in some particular items, the *line chart view* will show the corresponding time series curves (Fig. 2(d)). The *summary view*, *statistical view*, and *line chart view* complement one another and provide an effective way to explore the ranking changes at different levels of details.

4 Visualization Design

In this section, we describe in detail how RankExplorer visually illustrates ranking changes in a large time series dataset.

4.1 Design Rationale

The purpose of this work is to understand both value and ranking changes in time series data. One simple way is to leverage line charts. To achieve this goal, two sets of line charts are needed, and users have to check two views back and forth to find the potential correlation. In addition, line charts are only good at showing the change of a few data items over time. If there are thousands of curves, the change patterns will be difficult to recognize.

To remedy this, we augment a well-established time series data visualization, ThemeRiver. It plots each data item as a layer, with the width of the layer at a time point encodes a quantitative value of the data item at that time. It can show the changes of multiple data items at both the individual level and aggregation levels. However, ThemeRiver has two disadvantages: a) it often suffers from severe visual clutter for large datasets with thousands of items; b) it only encodes the change of one quantitative value. On the other hand, in our application, we are interested in both value and ranking changes of large time series data. To address the first issue, we introduce a segmentation method that partitions the data items into a controllable number of categories. For the second issue, we enhance the ThemeRiver visualization with color bars and changing glyphs to convey multiple aspects of ranking changes over time.

4.2 Segmentation

To provide a useful summarization with better visual clues, the segmentation in RankExplorer needs to meet the following two criteria:

C1: The average height of each layer should be similar. If one layer is too wide, it will give users an indication that this layer is more important than others, which is not always right; if one layer is too thin, users cannot see the color bars and changing glyphs clearly, which would hinder information understanding and the exploration process.

C2: The outer changes should be as small as possible. The relative independence of a category will help users focus on one particular layer with little distraction from others.

Next we discuss the mathematical formulation of the segmentation based on the above two criteria. To describe the segmentation method precisely, we use the following notations (given m time series v_1 to v_m, each has n values corresponding to time t_1 to t_n, respectively):

$r_i[k]$: the index of $v \in \{v_1, v_2, \ldots, v_m\}$ whose ranking is k at time t_i;
$e_i = (r_i[1], r_i[2], \ldots, r_i[m])$: the ranking vector of all the time series at time t_i;
$w(r_i[j])$: the value of $v_{r_i[j]}$ at time t_i;
$G = \{[g_1, g_2], [g_2+1, g_3], \ldots, [g_{l-1}+1, g_l]\}$: a segmentation scheme that segments the time series data into l groups, where $l \leq m$, $g_{i-1} \leq g_i$, $g_1 = 1$, and $g_l = m$;
$\mathrm{diff}(g_{k-1}, g_k, i)$: the number of items that are in the ranking range of $[g_{k-1}, g_k]$ of e_{i-1}, but not in the same range of e_i.

Mathematically, regarding a segmentation G, the first criterion can be expressed as:

$$f_1(k) = \frac{|W(g_{k-1}, g_k) - W(g_1, g_l)/m|}{W(g_1, g_l)/m} \tag{1}$$

where

$$W(g_{k-1}, g_k) = \sum_{i=1}^{n} \sum_{j=g_{k-1}}^{g_k} w(r_i[j]) \tag{2}$$

The second criterion can be formulated as:

$$f_2(k) = \frac{\mathrm{DIFF}(g_{k-1}, g_k)}{(n-1)(g_k - g_{k-1} + 1)} \tag{3}$$

Algorithm 1: A greedy approach for segmentation

Data: $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_t, n, m, \alpha$
Result: G
begin
  $G[1] = 1$; $start = 1$;
  for $i = 2$ *to* $m-1$ **do**
    $cost = MAXIMUM$; $G[i] = 1$;
    for $j = start$ *to* n **do**
      $temp = (1-\alpha)\frac{|W(j,start)-W(1,n)/m|}{W(1,n)/m} + \alpha\frac{\mathrm{Diff}(j,start)}{(t-1)(start-j+1)}$;
      if $cost > temp$ **then**
        $cost = temp$;
        $G[i] = j$;
    $start = G[i]$;

where

$$\mathrm{DIFF}(g_{k-1}, g_k) = \sum_{i=2}^{n} \mathrm{diff}(g_{k-1}, g_k, i) \tag{4}$$

Combine these two criteria together, the segmentation method is to minimize the following cost function:

$$f_c = \sum_{k=2}^{l} ((1-\alpha) * f_1(k) + \alpha * f_2(k)) \tag{5}$$

To enable smooth interaction, we adopt a greedy approach (Algorithm 1) to find an approximate result, which has time complexity $O(nml)$. Since the optimization method is less than perfect and users may have different needs, we also provide the following two operations to allow the user to interactively tune the segmentation results.

Tuning α In our implementation, parameter α is used to balance the two criteria of segmentation. Accordingly, we allow users to interactively tune this parameter.

Iterative Segmentation The segmentation is an iterative process. When a user is interested in one layer, s/he can double click this layer and the segmentation will be applied to this layer and a new RankExplorer visualization is generated for further exploration.

4.3 Summary View

4.3.1 Encoding Scheme for Trend Curves

To provide an overall ranking change of the select categories over time, we design a trend curve (at the top of Fig. 1). The height at each time epoch encodes the normalized degree of the ranking change (0 means there is no change of the ranking order, while 1 means the ranking order has totally changed).

A good measure of ranking change needs to meet the following requirements: 1) relative ranking changes between data items are considered; 2) a data item with a larger ranking change contributes more to the total ranking change; 3) appearance of a new data item and disappearance of an existing item also contributes the ranking change degree.

It is clear that all three requirements are closely related to the sortedness analysis of a sequence with respect to an ascending or descending order, which generally has three metrics to measure[1]: 1) the minimum number of items which can be deleted from the sequence to make the remaining ones fully sorted; 2) the smallest number of exchanges needed to sort the sequence; 3) the number of inversions which are the unsorted pairs in the sequence [25]. Among the three metrics, the inversion number can perfectly fulfil the first two requirements, and can be easily extended to fulfil the last requirement. As a result, we extend the concept of inversion number to measure the degree of ranking change.

[1] http://en.wikipedia.org/wiki/Inversion_(discrete_mathematics)

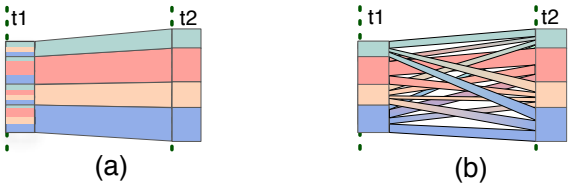

Fig. 3. Comparisons of two design alternatives to present the flowing-out items: a) using a color bar; b) using a flow map or bipartite graph.

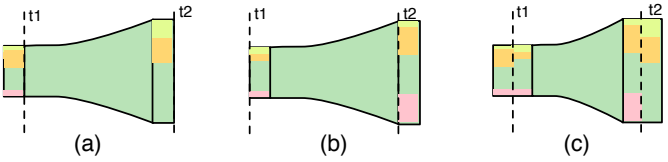

Fig. 4. Three design choices for the color bar in one cell: (a) only the flowing-in bar; (b) only the flowing-out bar; (c) both.

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a sequence of n distinct numbers, $B = \{b_1, \ldots, b_n\}$ be a permutation of A, and $P(x)$ be the permutation function such that $b_{P(i)} = a_i$. Then the inversion number [10] of B to A can be defined as:

$$Inv_A(B) = |\{(b_{P(i)}, b_{P(j)}) | i > j \,\text{and}\, P(i) < P(j), \forall 1 \le i, j \le n\}|$$

To meet the third requirement, we need extend the definition to handle A and B containing different items. To achieve this, we introduce three operators: $A \ominus B$ denoting the sub-sequence of A which removes all of the common elements in both A and B; $A \oplus B$ denoting the sequence of A with B appended to the end of A; and A^{-1} denoting the inverted sequence of A. Thus, we can transform A and B into two new sequences containing the same elements: $A' = A \oplus (B \ominus A)^{-1}$ and $B' = B \oplus (A \ominus B)^{-1}$ then calculate $Inv_{A'}(B')$.

In our system, for two neighboring time points i and $i+1$, we can compute the inversion number for the ranking sequences e_i and e_{i+1}. As the length of e'_i and e'_{i+1} may be as long as $2|e_i|$, we normalize the inversion number by dividing it by $\binom{2|e_i|}{2}$.

4.3.2 Encoding Scheme for Color Bars

In our application, many tasks require examining both the value and ranking changes. However, ThemeRiver can only encode one change at a time. Although we can use two ThemeRivers side by side to show them respectively, it is still very difficult to establish the correspondence and detect correlation between them. Thus, we prefer an integrated view, which can show both values and rankings changes simultaneously.

Another design option is to embed a flow map into the ThemeRiver. The content changes between all categories at two neighboring time points can be represented as a bipartite graph (Fig. 3(b)). However, this design may cause many line crossings that lead to visual clutter and even obscure the boundaries between categories.

After investigating various design alternatives, we pick a solution that embeds color bars into the ThemeRiver (Fig. 4), which provides three advantages in our applications: 1) it avoids line crossings; 2) it gives a statistical summary of the content change for each category at each time point; 3) it provides an integration to help users detect potential correlations between value and ranking changes.

There are two color bars for each layer at each time point (called cell for short): the flowing-in bar (Fig. 4(a)) and the flowing-out bar (Fig. 4(b)). The flowing-in bar encodes the layers of items in the current cell coming from the previous time point, while the flowing-out bar encodes layers they go to the next time point. To informatively show the changes, each color bar is divided into several color segments

with their colors encoding the sources or destinations of the changing items and their heights encoding the contributions of the changing items to the current cell. In addition, the color segments are stacked in accordance with the order of the layers, so that users can easily track and compare the color bars.

However, there are three special colors in color bars that we need to emphasize: 1) for the items that first appear in the ThemeRiver, we use white color to encode them; 2) for the items that reoccur in the ThemeRiver, we use dark green to encode them; 3) when zooming into a layer to see the sub-segmentations, we use dark red to encode the items coming from the upper layers and use dark green for the content coming from the lower layers.

Using our color bar design, it should not be surprising to see the major part of a color bar is filled by the color of the current layer or the colors of the neighboring layers. However, it is the content changes that happen between two distant layers, which is more valuable to users, because they are likely to indicate interesting patterns. To help users discover such patterns, we enhance the color bars by providing two interactions:

Non-Linear Scaling. In our color bar design, we scale the heights of segments in each color bar according to the distance between layers. The users can also interactively tune the scaling parameter, which is denoted by β. If β is bigger, the color bar gives more weight to segments representing changes from more distant layers. In particular:
$\beta = 1$: the color bar does not distort any segment in it;
$\beta = 0$: the color bar allocates all available space to the segment representing the current layer.

For a color bar in the i^{th} layer, we define the scaled height of the j^{th} segment in it as:

$$H(j) = \begin{cases} H_r(j) \times (1 + |i-j| \times (\beta - 1)), \beta \geq 1 \\ \frac{H_r(j) \times (i-j)^2 \times \beta}{\sqrt{1+((i-j)^4-1)\beta^2}}, 0 \leq \beta < 1 \end{cases} \quad (6)$$

where $H_r(j)$ and $H(j)$ represent the height of the j^{th} segment before and after scaling, respectively. Fig. 5 shows the color bars with the same information but in different values of scale parameter β.

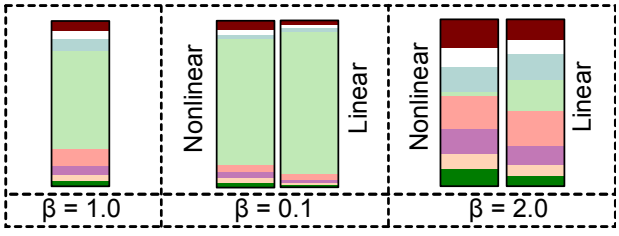

Fig. 5. A comparison between non-linear scaling and linear scaling. The color for the current layer is green. When $\beta = 1$, the color bar is non-distorted. When $\beta = 0.1$, the color segments with the smaller distance (pink and sky blue) shrink a lot for both non-linear and linear scaling, but the color segments at both ends change less in non-linear scaling than in linear scaling. When $\beta = 2.0$, the color segments at both ends scale more in non-linear scaling than in linear scaling.

Filtering. By default, we show all the color bars associated with all cells. However, when the changes become larger, users may not be able to easily see which cell changes the most. Thus, a filtering function is provided. Users can set a threshold to hide all the colors bars that are below the threshold.

4.3.3 Encoding Scheme for Changing Glyphs

Compared with color bars representing the content changes between layers, changing glyphs are used to reveal the changes within a single layer. The encoding scheme of changing glyphs is illustrated in Fig. 6. For two neighboring cells, we first extract all the data items contained in both cells. Then, for each data item, the ranking change is computed. A bar-chart-like design is used here to visually summarize all

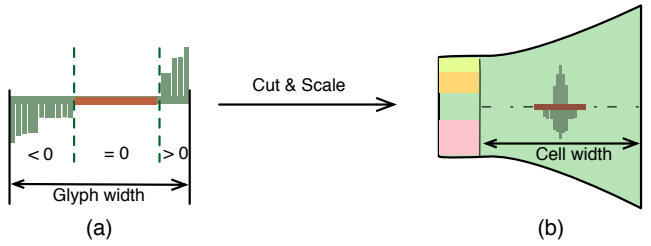

Fig. 6. The encoding scheme for changing glyphs. We divide the data items into three parts: the decreased part, the unchanged part, and the increased part and then pack them together.

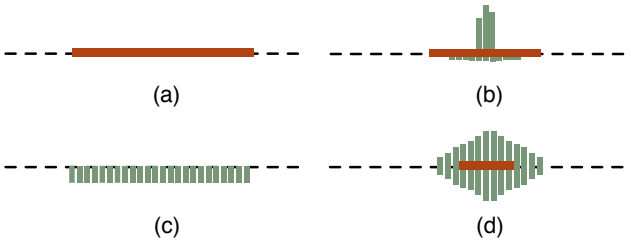

Fig. 7. Four different patterns we can observe from changing glyphs: (a) no change; (b) dramatic ranking increases of a few data items which cause others' rankings to decrease; (c) all rankings dropping equally due to some data items below the current layer going up to the layer above; (d) dramatic changes without clear pattern.

changes. In the changing glyph, a shared data item is represented by a vertical bar with its height encoding the change value. In addition, all the bars are sorted in increasing order (from negative to positive), while the width of the total bars encodes the percentage of the shared data items among all the data items at current time point (Fig. 6(a)). For example, if the percentage is 100%, the total width is equal to the width of the cell. In order to save space and facilitate comparison, we stack the unchanged part, increased part and the decreased part together, and make the shape symmetric (Fig. 6(b)).

By applying the encoding scheme to real datasets, users can quickly know what is the change between two time points in a layer. Fig. 7 shows four examples representing four major patterns.

4.4 Statistical View and Line Chart View

Although the *summary view* provides a nice visual summary by integrating trend curves, a ThemeRiver visualization, color bars, and changing glyphs together, more detailed information is needed for further investigation. Therefore, we design the *statistical view* and the *line chart view* to help users explain the patterns they discovered in the *summary view*.

After a user selects a cell, the *statistical view* will be updated to show more detailed information for the flowing-in and flowing-out items. As an example, in Fig. 10, each bar segment represents the number of items whose increased/decreased ranking changes are in the particular range. For example, the top left segment in Fig. 10 indicates the ranking of query "learning toys" increases by a number between 1079 and 1137. The colors in the *statistical view* are used to encode whether the data item(s) falling into that particular range have positive change (green) or negative change (red).

The *line chart view* is offered to see the raw data. When users find something interesting in the *summary view* or in the *statistical view*, they can choose the specific data item(s), which will then be shown in this view.

5 Case Studies

In this section, we illustrate how our system can be used to analyze time series data by applying it to two datasets: Bing search query data and US Fortune 500 data.

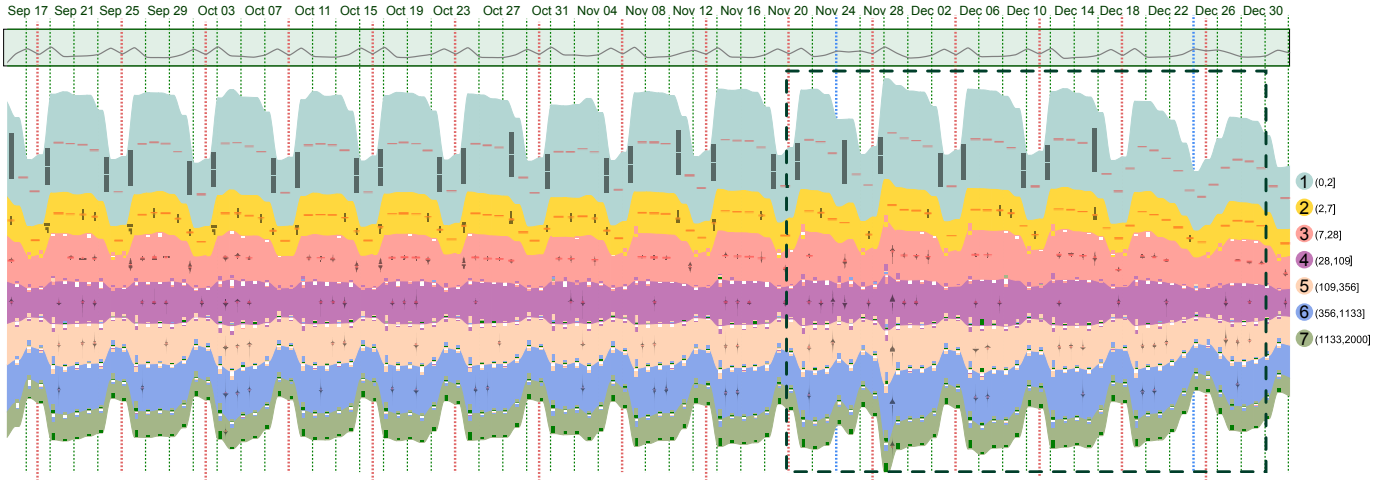

Fig. 8. The Bing search query data from Sep. 15 to Dec. 29 in 2011. The periodic pattern is quite clear. The Sundays are marked as red dotted lines, while Thanksgiving Day and Christmas Eve are marked as blue dotted lines.

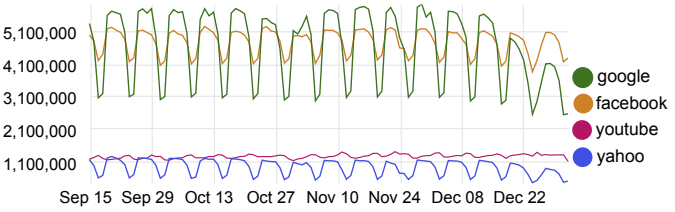

Fig. 9. Time series curves for the top 4 hottest queries. The rank increase of "facebook" is due to the decreasing of the query count of "google". The ranking of "youtube" is quite stable after Oct. 17, remaining in the third place.

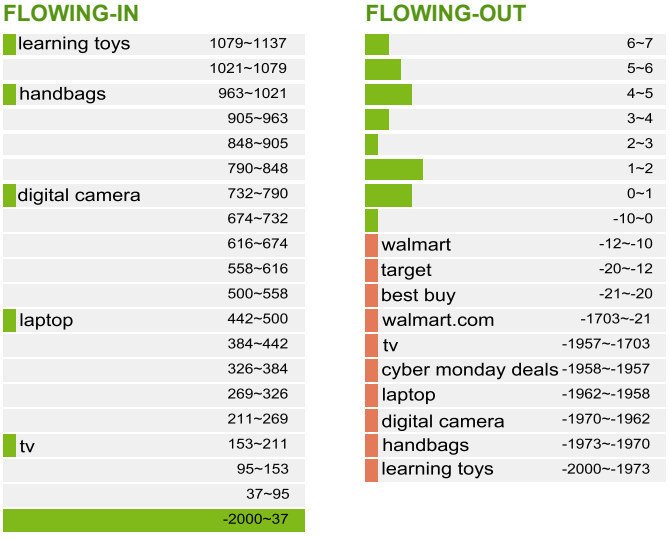

Fig. 10. An example of statistic view. This figure shows the statistical information of the cell in layer 3 on Nov. 28 (region B in Fig. 1). The length of each bar represents the number of distinct queries whose increased/decreased ranking changes are in the particular range. The queries in this figure are manually labelled.

5.1 Case Study 1: Bing Search Query Data

We collect the total Bing search query log data from Sep. 15 to Dec. 29 in 2011. The task is to explore the top 2000 queries over time. We segment the data by ranking and apply our system to the data. For each cell, it contains several queries, while the height of the cell encodes the total query count of these queries. Due to the limitation that people can only efficiently distinguish a dozen colors [33], we segment the data into seven groups. The result is shown in Fig. 8. At first look, we can clearly see that the height of the ThemeRiver changes periodically. Generally, on weekends and holidays (Thanksgiving and Christmas), the total query count is significantly smaller than the ones on weekdays. Also, we can find some patterns from the trend curve, which shows the degree of ranking changes for the whole time series curves: on each Saturday and Monday, the ranking change is larger; on Nov. 28, the change is larger, too.

Then, we explore the data at layer level. In layer 1, there are only two queries: "google" and "facebook". These two are always the top two as the color bar is always the same color. The change in height of this layer follows the same pattern as the change in height of the ThemeRiver. From the changing glyphs, we can see that the rankings of these two queries change on weekends: on weekdays, "google" is above "facebook", while on weekends and holidays, their order is reversed. Looking into the *line chart view* (Fig. 9), we can see the reason for the ranking change. It is not because of the increasing query count for "facebook", but the decreasing query count for "google".

For further exploration, we target at the time period from Nov. 20 to Dec. 29 in 2011 as two interesting patterns appear in this period. One is Nov. 28 when the height of the ThemeRiver is the largest, and another is around Dec. 25 due to Christmas. When we explore the data in this time period, first we can see that the segmentation has changed (Fig. 1). Layer 2 now only contains one query "youtube" and the ranking is very stable (at the third place). Also, the height of this layer is stable too, which means, unlike "google" and "facebook", the query count of "youtube" does not change much (Fig. 9). From the trend curve, it is obvious that the degree of ranking changes of layer 4 (the purple curve) is much larger than the global degree of ranking changes (the grey curve).

By tuning the scale parameter of the color bar, an outlier appears in layer 4. From the flowing-in bar, we can clearly see that on Nov. 28, the cell contains a lot of items that come from layers 6 and 7 (region B in Fig. 1). Also, an item in this cell goes into layer 3, which is "amazon". By clicking this cell, we can see the details in the *statistical view* shown in Fig. 10.

From the *statistical view*, we can see that there are five queries coming into the cell with a big increase in rank, which are "learning toys", "handbags", "digital camera", "laptop", and "tv". In addition, as indicated by the flowing-out part, there are ten queries that decrease a lot in the next day, which are "walmart", "walmart.com", "target", "best buy" and "cyber monday deal" together with the five queries men-

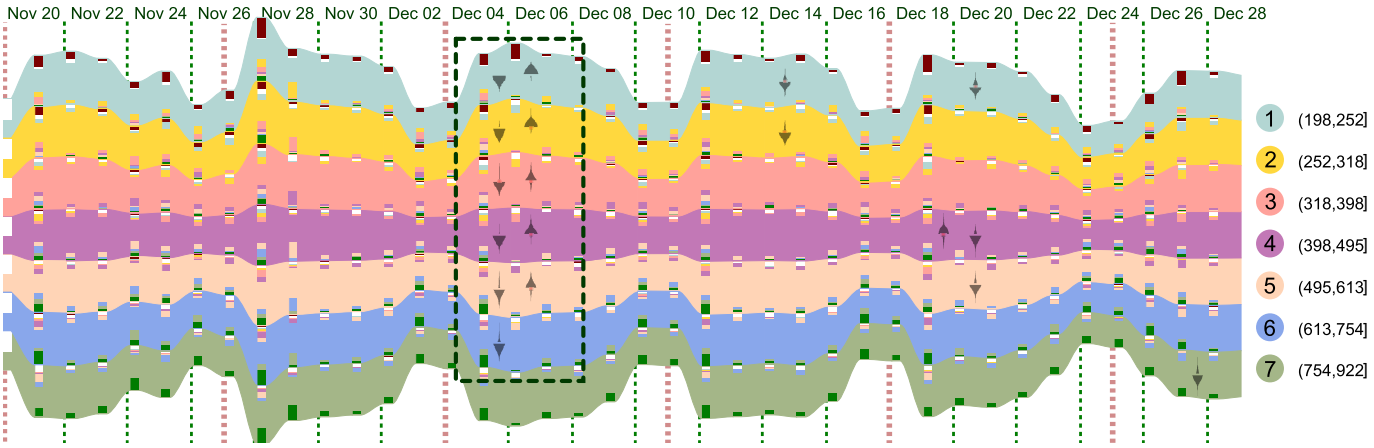

Fig. 11. The segmentation result for the layer 6 in Fig. 1 for further exploration. In the dashed rectangle, we can see the pattern that the rankings of the items in all of the cell of Dec. 5 first decrease in Dec.6 and then increase in Dec. 7, which indicates that several queries suddenly appear in upper layers in Dec. 6.

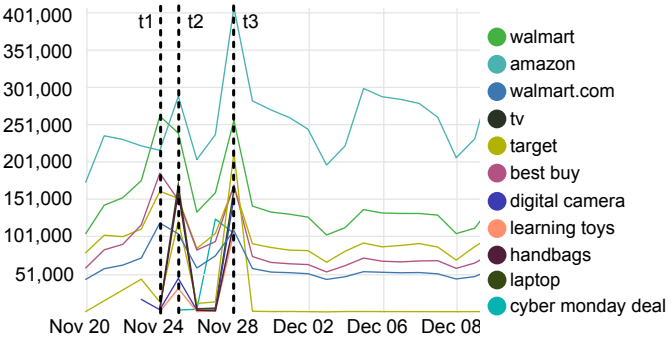

Fig. 12. Time series curves of selected queries. At t1 (Nov. 24), "walmart", "walmart.com", "target", and "best buy" reach the first peaks. At t2 (Nov. 25), "learning toys", "handbags", "digital camera", "laptop", "tv", and "amazon" reach the first peaks. At t3 (Nov. 28), all of them reach the second peaks.

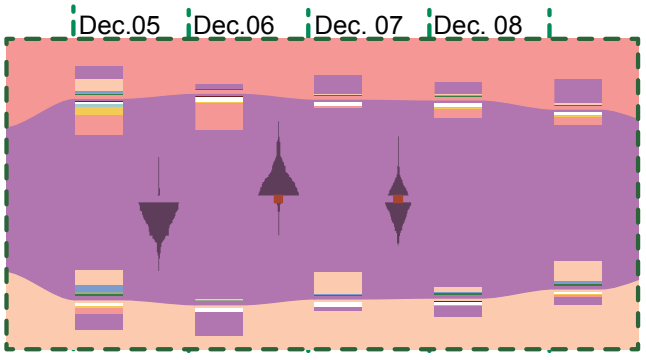

Fig. 13. The zoom-in view of the dashed rectangle in Fig. 11 to show the cells in layer 4.

tioned above the next day. We then choose all of the queries mentioned and their curves are plotted in the *line chart view* (Fig. 12).

From the *line chart view*, we can see that all of the queries reach their peaks on Nov. 28, which is Cyber Monday. However, four queries, "walmart", "walmart.com", "best buy", reach their first peaks on Nov. 24; the other queries, "learning toys", "handbags", "digital camera", "laptop", "tv", and "amazon" reach the first peaks on Nov. 25, which is the Black Friday.

Interestingly, in layer 4 of Fig. 1, there are several cells where white color segments occurs, which means some new queries first appear in the top 2000 list. On Nov. 21, "Kutcher $290 million court battle" and "brittney jones on ashton kutcher divorce" appear. On Dec. 8 "hoover dam helicopter crash" and "lindsay lohan spread" appear. On Dec. 9, "angelina jolie wardrobe malfunction" appear. On Dec. 18, "duggars stillborn photo" and "kobe divorce vanessa bryant" appear. In all, those newly appeared queries are about celebrity gossip, except for "hoover dam helicopter crash".

On Christmas Eve, the total query count of all the top 2000 queries is the lowest during the whole month. However, in layer 4, there is still a color segment indicating that the rankings of some queries in layer 6 increase and these queries come into layer 4. These queries are "norad santa tracker 2011" and "santa tracker".

We then zoom into layer 6 by double clicking, and the result is shown in Fig. 11. By thread filtering operation and color bar tuning, we can clearly see in the dashed rectangle, the glyphs show a strong shifting pattern: the rankings of the items in all of the cells of Dec. 5 decrease on Dec. 6, and then increase on Dec. 7, which indicates several queries suddenly appear in upper layers only once on Dec. 6. From the color bar (Fig. 13), we can also see the shift pattern. We go back to check Fig. 1 and find that in layer 5, a lot of new queries first appear on Dec. 6 and then disappear on Dec. 7 (region D), which causes the shift.

5.2 Case Study 2: US Fortune 500 Data

We collect US Fortune 500 Data from 1955 to 2010. This data contains the top 500 companies in US with their revenues each year. Also, we use the ranking to segment the data and use the revenue as the aggregation value. The result is shown in Fig. 14.

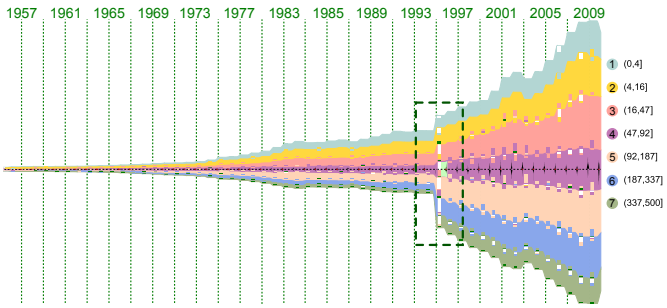

Fig. 14. US Fortune 500 data from 1955 to 2010. There is a great change in 1995 (in the dashed rectangle).

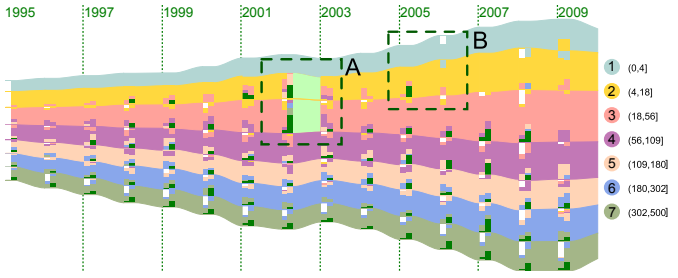

Fig. 15. US Fortune 500 data from 1995 to 2010. In region A, there is a large dark green segment in the flowing-out bar, which means there are some companies dropping out of the top 500 in 2003; In region B, there is a white segment in the flowing-in bar, which means there are some companies suddenly that appear in the top 500.

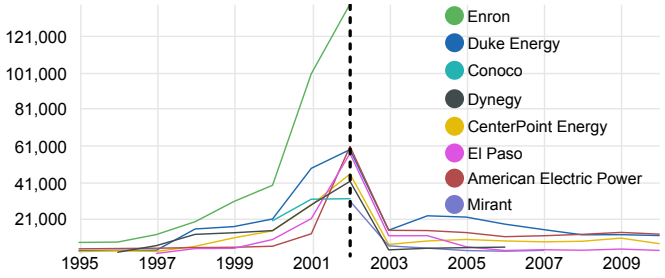

Fig. 16. Time series curves for the selected energy companies. They reach the peaks in 2002 and drop a lot in 2003. The highest one is "Enron", which filed bankruptcy protection in 2003.

We can clearly see that there is a big change in 1995: the height changes a lot. In addition, in each cell, there are a lot of new items. By searching some related background information, we know that the original US Fortune 500 (before 1995) consists of only the largest publicly held industrial companies, and the list of the 500 largest service-oriented companies were combined together. Thus we focus on the data from 1995 to 2010 for further exploration.

By only showing the flowing-in color bar, there is no cell with strong visual patterns. Thus, we show both flowing-in and flowing-out color bars, and find, in 2002, both layers 2 and 3 stand out (region A in Fig. 15). From the flowing-out bar, we can see there is a large dark green segment, which indicates several companies flow out from layers 2 and 3. Thereby, their rankings would be out of the top 500. After selecting these two cells and exploring the *statistical view*, we can see there are twelve companies with their rankings decreased by more than 100. Among them there are eight companies in the energy field: "Dynegy", "El Paso", "Mirant", "CenterPoint Energy", "Enron", "Duke Energy", "American Electric Power" and "Conoco". By plotting them in the *line chart view*, we can get Fig. 16.

From the line chart, we can see that, for all companies, their revenues reach the highest points in 2002. However, after that, they either decrease significantly or disappear. Especially for "Enron", we search its history and find that it filed for bankruptcy protection due to the "Enron scandal" in 2002, which could be the reason for the common ranking decrease of energy companies.

In 2005, a white segment appears in the flowing-in bar of layer 1, which shows there is a new company, which is "Chevron" (region B in Fig. 15). By tracing back to 2004, there is a dark green segment in the flowing-out bar in layer 2, which means there is a company moving out of top 500. It turns out to be "ChevronTexaco". Interestingly, the change happens because of the renaming of the company as indicated by Chevron's history.

6 Comparative Study

To further demonstrate the effectiveness of RankExplorer, we conducted a comparative study with a base line system. The baseline system contains four line charts, which illustrate four types of statistics on two levels. The first level shows the dataset related statistics, including the trend curves for the whole dataset and each category (Line Chart 1 in Fig. 17), as well as the aggregated outer changes for each category (Line Chart 2 in Fig. 17). The second level (Line Charts 3 and 4 in Fig. 17) related statistics for a selected category, including the flowing-in percentage from every category (which is similar to the color bar) and the statistics of inner changes inside a given category over time (which is similar to the changing glyph).

We recruited ten users who had never used RankExplorer. All participants are experienced computers users. At the beginning of each user session, we gave a brief tutorial of both systems. The data we used was the top 300 queries from Nov. 20 to Dec. 29. Then the participants were asked to finish five tasks. Considering the complexity involved and time required in performing the tasks, the following simple tasks were selected:

- **T1**: Find the time points with the largest degree of ranking changes globally;
- **T2**: Find the time points with the largest degree of ranking changes in a category;
- **T3**: Find the time points in a category when the rankings of most queries increased;
- **T4**: In the given categories, find the time points with new queries;
- **T5**: Find the time points in the given categories with one specific changing pattern: there is no incoming queries from other categories and only two queries swap their rankings.

All participants complete the tasks correctly. However, the answer time (the time spent on each question) is significantly different for different tasks with different systems. **T1** and **T2** can be finished by observing the trend curve. Since it is provided in both systems, the answer time is similar. However, tasks **T3**, **T4**, and **T5**, require users to examine more details. Accordingly, the answer time by using the baseline system is longer than the one of RankExplorer, ranging from 1.1 times to 5.6 times.

Specifically, we take **T5** (the most complicated task of the five) as an example to illustrate the difference. As shown in Fig. 17, by RankExplorer, the participants only need to find the cells, in which the changing glyph contains only two bars (one below and the other above the red line) having the same height. Meanwhile, the color bar on the right only contains the color representing the current category (region A in Fig. 17). However, in the baseline system, the participants had to select all categories one by one, then examine the value of each time point in Line Chart 3 (region B in Fig. 17). For each satisfied time point, they had to further explore Line Chart 4 to check whether there is only one query increased and one query decreased. (region C in Fig. 17). Such exploration is often repeated several times to find the desire patterns, which is very time-consuming.

The user survey also presented five subjective measures for RankExplorer. All the subjective measures were rated on a 5-point scale: Strongly Agree (++), Slightly Agree (+), Neutral (O), Slightly Disagree (–), Strongly Disagree (– –). The result is shown as follows:

	++	+	O	–	– –
RankExplore is easy to understand	4	4	2		
RankExplore is easy to use	5	4	1		
RankExplore is useful	4	6			
Color bar is useful	7	2	1		
Changing glyph is useful	5	4	1		

Through the discussions with the participants, we found that they preferred RankExplorer for two reasons. First, it allows users to examine ranking changes at different levels of detail, from the overall ranking/value changes to the individual ranking/value changes. Second, the color bar and changing glyph are integrated together to help users better track both the inner/outer changes at the same time. For example, one participant commented "*The color bar together with the changing glyph helps me recognize the changing patterns quickly. However, it is quite hard to do the same things in the baseline system, especially when patterns get more complicated*". In addition, all the

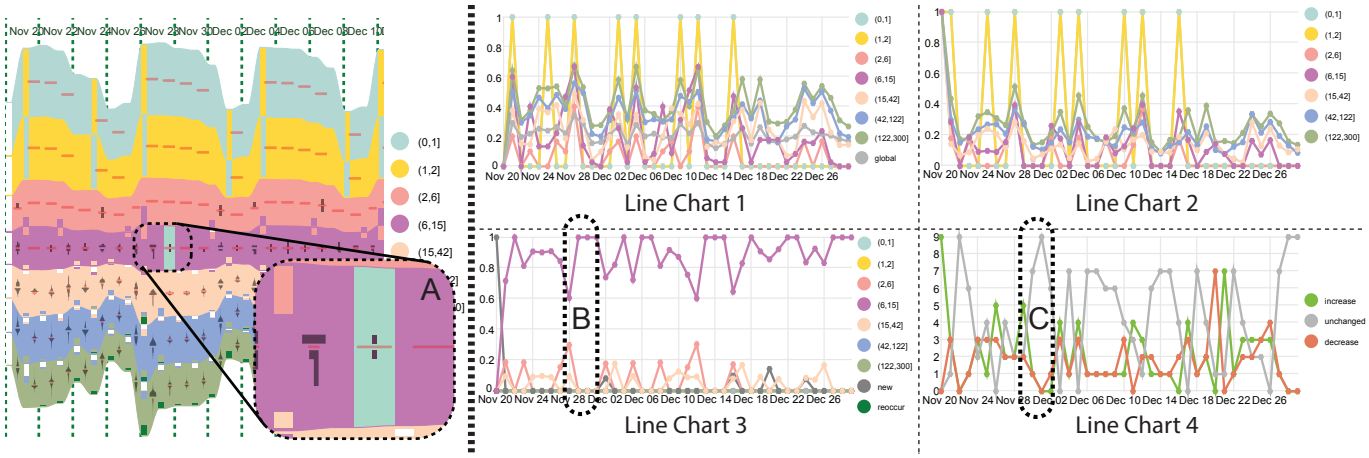

Fig. 17. Illustration of different ways to finish **T5**. In region A, the glyph shows there are only two queries with rankings swapped and the color bar on the right contains only one purple segment (the color of the current category), which represents there is no query from other categories. The same pattern can only be detected in the baseline system by exploring the regions B and C simultaneously.

participants complained about the visual clutter in the lines charts. One of them stated "*The curves in the line chart are very messy; it's really hard for me to differentiate them at the first look*". The participants also suggested a few potential improvements. Five out of ten users expressed the need to explore the potential correlations between different ranking change patterns. For example, one participant said: "*It's very interesting to detect so many useful ranking change patterns, but I am also interested in knowing the correlations among these patterns. For example, I want to know which pattern leads to this (pointing to one layer cell)*". Two out of ten participants stated that they disliked RankExporer's inability in supporting more categories.

7 Discussions

The case studies and the comparative study clearly demonstrate that our visualization method can effectively reveal the value and ranking changes in large time series data. The design of trend curves, color bars and changing glyphs indeed can show the changes at different levels of detail. For example, in the first case study, we applied our system to a dataset containing 2000 data items and detected several interesting patterns. In this dataset, an iterative segmentation method is applied to handle a large number of data items at different levels of details. Accordingly, this segmentation method enables RankExplorer to handle huge amounts of data.

However, our method also suffers from several limitations. First, as the color is used in the color bar to encode the ranking category, the method can show less than ten categories in one view. For large data, users have to use level-of-detail or "overview first, filter and zoom, and details on demand" to interactively explore the whole data. Second, for turbulent changes, it is possible that seven or eight different color segments are squeezed into one color bar and such color bars are everywhere in the ThemeRiver, which causes visual clutter to some extent. However, the RankExplorer visualization can still provide users a quick overview of the dramatic changes between layers. They could zoom in to see the details of the color bars. Based on our observations, in many applications, rankings have some kind of stability and it is unlikely that the rankings will change randomly. In addition, we can filter out some color bars based on the specific task requirements in an application. For example, we can ignore small changes and filter out all the color bars indicating only minor changes. Also, the filtering operation can be also applied to the changing glyphs.

8 Conclusion and Future Work

In this paper, we have presented RankExplorer, a novel visualization method to help users explore both the value and ranking changes in large time series data. Our system extends ThemeRiver with color bars and changing glyphs to provide a level-of-detail view for both value and ranking changes. The design rational, the encoding schemes, and the case studies have been presented. Compared with other visualization methods for time series curves, our design has some clear advantages. It keeps the advantages of ThemeRiver while important extensions are also made so that the changes of two values can be simultaneously revealed and potential correlations can be better detected. Our method is not limited to analyzing ranking changes in time series data. For data in which two values change simultaneously and users want to find their correlation, they can extend our method to use ThemeRiver to encode the changes of one value over time while embed the color bars and line charts to summarize the changes of another value.

There are several avenues for future work. There might be uncertainties and errors in the data, which our current system does not support. It is an interesting topic to explore and certainly our next step to study uncertainty analysis to handle such data. In addition, there might be complicated correlations for ranking changes. For example, the ranking change of one item may cause the ranking changes of others, or the change pattern of one item over a time period is similar to the change pattern of another item over another time period. We plan to integrate these analysis methods into our system to enable users to better detect such correlations. Furthermore, our system faces the visual clutter problem if the changes are too dramatic between layers. We will investigate more effective clutter reduction methods for this problem in future work.

Acknowledgments

The authors wish to thank Yangqiu Song for offering the data, the participants for doing comparative study, and the anonymous reviewers for their useful comments. This work was supported in part by grant HK RGC GRF 619309 and a grant from Microsoft Research Asia.

References

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.

[2] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–52, 2008.

[3] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. J. Gao, X. Tong, and H. Qu. TextFlow: towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–21, 2011.

[4] C. Daassi, L. Nigay, and M. Fauvet. A taxonomy of temporal data visualization techniques. *Information-Interaction-Intelligence*, 5(2):41–63, 2005.

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

[6] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–23, 2007.

[7] M. Hao, U. Dayal, D. Keim, and T. Schreck. Importance-driven visualization layouts for large time series data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 203–210. IEEE, 2005.

[8] M. Hao, U. Dayal, D. Keim, and T. Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. *Symposium A Quarterly Journal In Modern Foreign Literatures*, pages 1–8, 2007.

[9] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: visualizing theme changes over time. *In Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 115–123, 2000.

[10] T. H.Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 3rd edition, Feb. 2009.

[11] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. *In Proceedings of the 27th International conference on Human factors in computing systems (CHI)*, pages 1303–1312, 2009.

[12] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[13] W. Javed and N. Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. *In IEEE Pacific Visualization Symposium (PacificVis)*, pages 33–40, Mar. 2010.

[14] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–34, 2010.

[15] D. Keim, M. Ankerst, and H. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the 6th conference on Visualization'95*, pages 279–286. IEEE Computer Society, 1995.

[16] D. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler. A Spectral Visualization System for Analyzing Financial Time Series Data. *Time*, 2006.

[17] P. Kidwell, G. Lebanon, and W. Cleveland. Visualizing incomplete and partially ranked data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1356–1363, 2008.

[18] R. Kincaid. Signallens: Focus+ context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, 2010.

[19] M. Krstajic, E. Bertini, and D. Keim. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432–2439, 2011.

[20] B. Lee, N. Riche, A. Karlson, and S. Carpendale. Sparkclouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, 2010.

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

[22] Z. Liu, J. Stasko, and T. Sullivan. Selltrend: Inter-attribute visual analysis of temporal transaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1025–1032, 2009.

[23] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: interactive visual exploration of system management time-series data. In *Proceeding of the 26th International conference on Human factors in computing systems (CHI)*, pages 1483–1492, 2008.

[24] W. Muller and H. Schumann. Visualization methods for time-dependent data-an overview. In *Proceedings of the Winter Simulation Conference*, volume 1, pages 737–745. IEEE, 2003.

[25] P. Mutzel and J. Michael. Simple and Efficient Bilayer Cross Counting. *Journal of Graph Algorithms and Applications*, 8(2):179–194, 2004.

[26] W. Playfair. *The Commercial and Political Atlas and Statistical Breviary*. New York: Cambridge Univeristy Press. (Original work published 1786), 2005.

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

[28] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL*, pages 336–343, 1996.

[29] S. Silva and T. Catarci. Visualization of linear time-oriented data: a survey. In *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, volume 1, pages 310–319. IEEE, 2000.

[30] J. Van Wijk and E. Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*, pages 4–9,. IEEE Comput. Soc, 1999.

[31] C. Wang, H. Yu, and K. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.

[32] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith. Temporal summaries: supporting temporal categorical searching, aggregation and comparison. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1049–56, 2009.

[33] C. Ware. *Information Visualization: Perception for Design (Interactive Technologies)*. Morgan Kaufmann, 1st edition, Feb. 2000.

[34] M. Wattenberg. Baby names, visualization, and social data analysis. *In Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 1–7, 2005.

[35] J. Woodring and H.-W. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–37, 2009.

[36] J. Zhao, F. Chevalier, and R. Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the 29th international conference on Human factors in computing systems (CHI)*, pages 1737–1746. ACM, 2011.

[37] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.

[38] H. Ziegler, M. Jenny, T. Gruse, and D. Keim. Visual market sector analysis for financial time series data. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 83–90, 2010.