Scholarly Paper: Interactive Visualizations for Comparing Two Trees With Structure and Node Value Changes

John Alexis Guerra Gomez
jguerrag@cs.umd.edu
Department of Computer Science[1]
Human-Computer Interaction Lab[2]
University of Maryland, College Park, MD

Under the supervision of:,
Catherine Plaisant[2] and Ben Shneiderman[1,2]
Department of Computer Science[1]
Human-Computer Interaction Lab[2]
University of Maryland, College Park, MD

And the collaboration of:
Audra Buck-Coleman
Department of Art
University of Maryland, College Park, MD

Abstract

A common data analysis task is to compare pairs of trees to detect changes in leaf or interior node values and to identify created and removed nodes. However even in trees with just a dozen nodes it is difficult to find those differences. We present TreeVersity, a new interactive visualization that gives users powerful tools to detect both node value changes and topological differences. TreeVersity uses dual comparison techniques (side-by-side and explicit differences) and tabular representations, to facilitate the understanding and navigation of the differences. TreeVersity's design employs carefully-designed color palettes to show positive/negative, absolute, and relative value changes; shapes that preattentively show these changes; and novel graphical approaches that highlight created and removed nodes. We illustrate TreeVersity's functionality through comparison of the 2011 and 2012 U.S. Federal Budget. Eight usability test participants, with no initial training, identified many differences between the two trees, while suggesting improvements which were implemented.
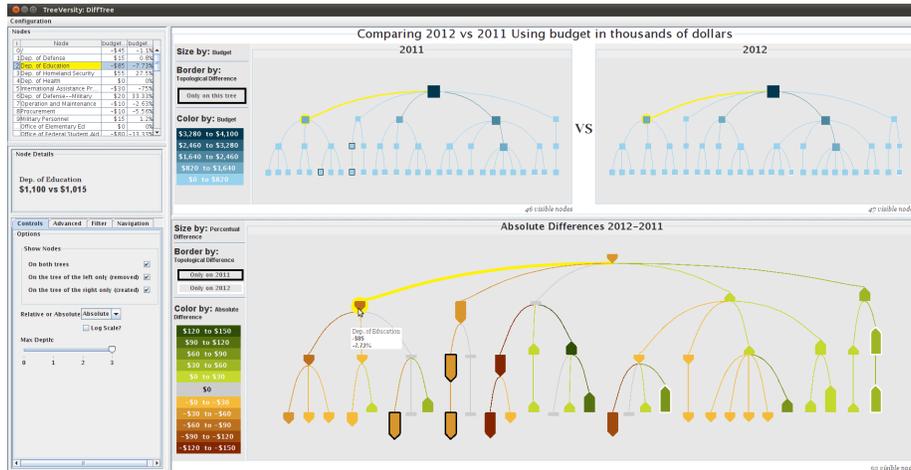
1

Figure 1: TreeVersity comparison interface. On the top are the two original trees being compared (budgets for 2011 and 2012). At the bottom the DiffTree shows the amount of change for each node. The glyph called "the bullet" points up to denote increases, and down for decreases. Nodes that have the same value in both trees are shown as small gray rectangles. The created and removed nodes are highlighted with a thick white or black border respectively. In this example the height of the Bullet is proportional to the relative change (in \%) while the color is mapped to the absolute change (in Dollars) making it easy to spot the changes that are significant in both absolute and relative terms, i.e. the dark tall bullets. Novice users can start with a redundant encoding using the same variable for both color and size.

INTRODUCTION

Hierarchies help us organize and understand information. Examples include the U.S. Federal Budget, the evolutionary tree of species, the hierarchies of basketball teams, business organizational charts. Much research has been done toward visualizing, navigating and understanding tree structures. Techniques such as node link representations [26], TreeMaps [17], Radial representations [7] and Icicle trees [19] are now often used in scientific and non-scientific publications.

Once one understands the data represented in a single tree, the next stage often is comparing trees. What are the big losers and winners in next year's budget proposal? How do the overall budgets of France and Germany compare? Have the salaries associated with different levels of responsibilities changed in the company since last year? The answers need to reveal both topological differences (e.g. what nodes appear, disappear or move), and node attribute value differences (increases and decreases). Biologists are interested in finding created, removed and relocated nodes when comparing taxonomies of species[24], while economists are mostly interested in the relative

change in the closing stock market prices [32]. While most related work has focused on one or the other aspects we propose a novel tree comparison tool able to addresses differences in both node values and changes in topology.

Our prototype - called TreeVersity (Figure 2) is designed to support four types of tree comparison problems:

Type 1   Positive and negative changes in leaf nodes' values with no changes in topology. Example: Comparing the stock market's closing prices between today and yesterday across a hierarchy of market sectors, assuming no stocks are created or deleted.

Type 2   Positive and negative changes in leaves and interior nodes' values with no changes in topology. Example: Comparing the salaries in an orgchart between two years, when no reorganization has occurred.

Type 3   Positive and negative changes in leaf nodes' values with changes in topology. Example: Comparing the budget of U.S. government between two years. Some agencies and departments have been created or terminated.

Type 4   Positive and negative changes in leaves and interior node's values, with changes in topology. Example: Comparing the number of page visits between two months using the website file hierarchy as a natural organization. Some pages might be created or removed, and each page in the hierarchy has an independent number of visits.

RELATED WORK

This section focuses on research that has been done on comparing, visualizing and analyzing multiple tree structures. There is substantial work on single tree structures, but since they are not relevant to the objective of comparison, it won't be described in this document. For a surveys of single tree visualizations please refer to [3, 15, 13, 27, 18].

The related work has been categorized in three areas according to each project's focus: topological comparison, node value comparisons and algorithmically oriented approaches.

Topological Comparison

Most of the tree comparison work has been done on comparing topological changes between tree structures. This might have been influenced by the well-known problem of comparing taxonomies of species. TreeJuxtaposer by Munzer et al.[24] is one of the best examples, presenting an efficient algorithm for comparing hierarchies. It uses a node link representation with side-by-side comparison and a focus+context technique
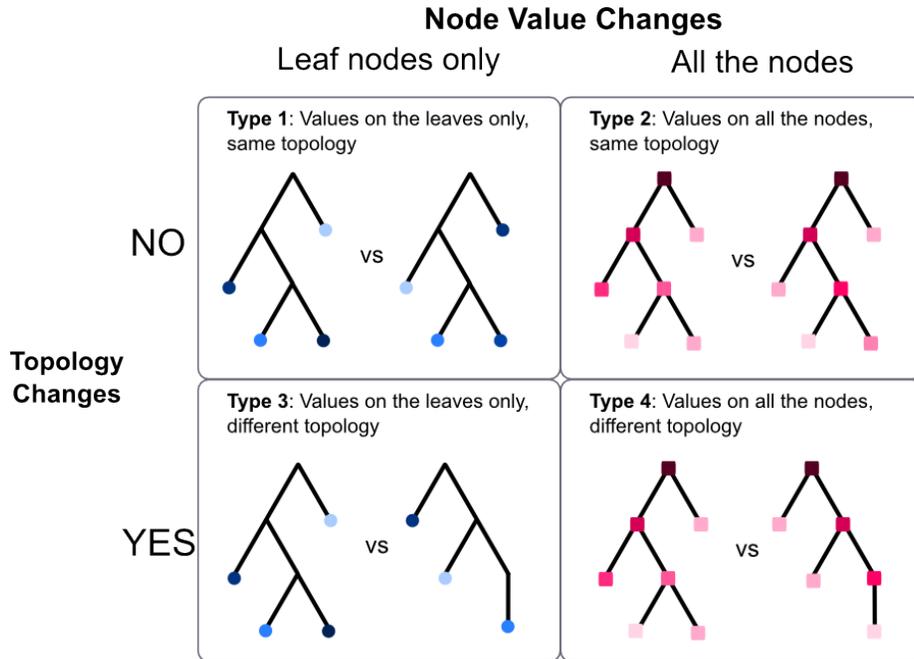
Figure 2: Types of tree comparison problems handled by TreeVersity

with guaranteed visibility. TreeJuxtaposer scales well with the number of nodes. Alternatively, Graham et al.'s [11] Icicle-like [19] representation and Bremm et al. [5] node-link visualization, scale to a large number of trees by dividing the screen space into small interconnected views of the compared trees, but are limited by the screen size. In later work [12] Graham addresses this by switching from the small multiples to an aggregated representation using directed acyclic graphs (DAG). Others have used the concept of aggregation of multiple trees in one view, starting with Furnas et al. [8] who proposed the concept in 1994 and CandidTree[20] that uses a node-link representation that uses color, shapes and dotted lines to represent uncertainty. Amenta and Klingner's TreeSet [1] takes a different approach to comparing a large number of taxonomies by calculating a bi-dimensional metric representing each tree and plotting them in a scatter plot. Card et al.'s TimeTree [6] explored the concept of time changing hierarchies, combining SpaceTrees [26] with timeline to analyze hierarchies that evolve with the time.

The InfoVis2003 contest [14] promoted the development of projects on topological tree comparison. Some of the winning submissions presented innovative solutions for the problem, such as TreeJuxtaposer [24], already described. Others include Zoomology [16] which used radial representations combined with zooming interfaces, Info-Zoom [28] which used condensed side-by-side tables, EVAT [2] with radial side-by-side comparisons and TaxoNote [23] with a condensed Microsoft Windows Explorer-

like representation. However, many of these promising projects did not evolve beyond the competition's two page submission requirement.

Finally other approaches use zooming interfaces such as MoireTrees [22], which allows navigation of multi hierarchies (different trees that categorize a shared group of leaf nodes) using zooming and radial displays, and DoubleTree [25], that uses two connected, side-by-side SpaceTrees [26] to highlight topological differences between taxonomies.

Despite the substantial work on topological differences between trees, to the best of our knowledge, none of these solutions addresses the problem of comparing changes in node values. TreeVersity takes the task of comparison tree structures one step further, by looking also at node value changes. However more complex topological comparison features already supported by these projects, like finding moved nodes and subtrees, have not yet been addressed in the TreeVersity design. More specifically, TreeVersity performs topological comparison by identifying created and removed nodes, reveals changes in the node values, tackling a richer set of problems than those that are restricted to topological differences only.

Node Values Comparison

The work on comparing node values is more limited, usually employing treemaps. The original treemap tool [17] allowed the display of changing values on the hierarchy but it was never developed for comparison. Animated TreeMaps [9] represented changes in the nodes' attribute values using animation, focusing on stabilizing the layout. Both projects rely on user's memory to keep track of the amount of change and the location of the nodes which can be taxing and confusing. TreeVersity in contrast combines side-by-side comparison with explicit differences visualizations that allow users to navigate differences in a more explicit way. SmartMoney's Map of the Market [32] represents stock market price changes using colored treemaps[1]. This approach has proven to be popular, however it only presents relative differences in the leaf nodes without topological changes, or what was called problem Type 1 in the introduction.

Finally, the Contrast TreeMaps [31] represents the only project that, to the best of our knowledge, combines some topological differences and changes in node values. It modified the traditional treemap technique by splitting each of the nodes' rectangular shapes into two triangles to represent value changes and structural differences, as seen in Figure 3. Compared to the Contrast TreeMaps, TreeVersity covers a broader set of problems (problems Type 3 and 4), uses a codification scheme that we believe is easier to comprehend, and uses a mixed representation approach that allows the navigation of the differences in a more direct way.

Algorithmically oriented

The final approach for tree comparison makes use of tree metrics, which usually are algorithms that calculate distances between two or more trees. These metrics can be classified by the type of comparison they make, and Bille [4] presents an excellent

---

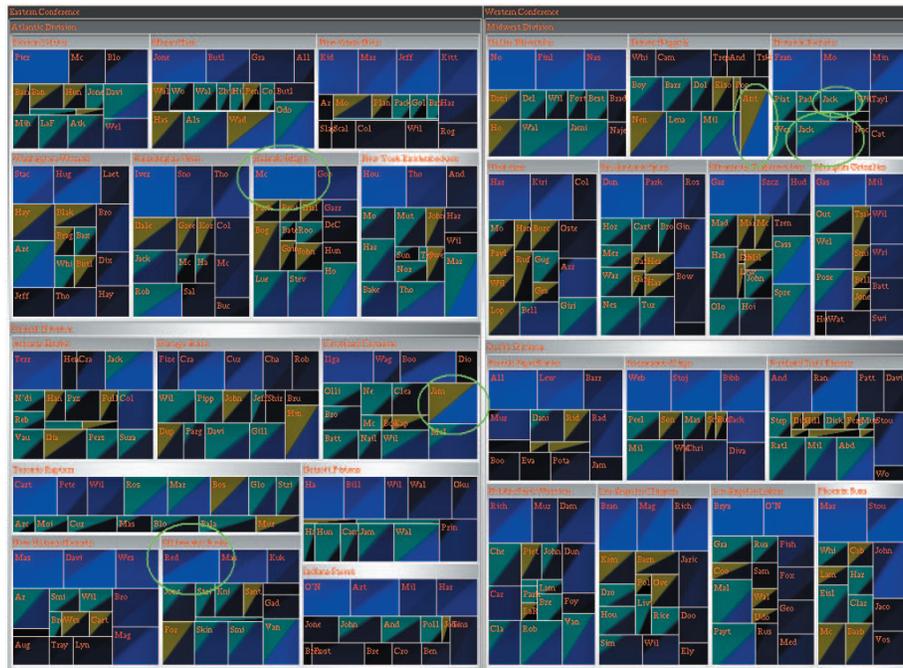[1]http://www.smartmoney.com/map-of-the-market/

Figure 3: Contrast TreeMaps [31] uses color and different shading techniques to encode node value and structural changes. The image shows the differences in NBA player's points per game between two seasons, categorized by teams and conferences. The paper [31] explains: "For an item, if both corners are in the blue to black range, the player was in the same team for both seasons. If the color for the 02-03 season is pine green, it means the player transferred to this team in the second season. If the color for the 02-03 season is dark yellow, the player joined the NBA in the second season".

survey of them. According to him the most important classes of metrics are Edit Distance, Alignment Distance and Inclusion (subtrees). In this work he describes efficient algorithms for each of this areas that could be used to compare many trees at once.

Another common related strategy for analyzing multiple trees is the consensus tree [1, 30, 21, 29]. This a technique used in Phylogenetic analysis for summarizing many trees into one. Future versions of TreeVersity will include Bille's metrics and the consensus tree strategy.

TREEVERSITY

Overall interface description

TreeVersity was designed to help users identify the following types of differences:

- Created and removed nodes.

- Absolute and relative differences of the node attributes values.

- Polarity of the differences.

- Changes in both the leaf nodes of the tree and the interior nodes.

- Variations between siblings and at different levels of the tree.

At this time the prototype does not support relocated nodes and subtrees.

According to Gleicher et al. [10], there are three common approaches used to compare data structures: side-by-side comparison (juxtaposition), superposition, and explicit difference (aggregation). TreeVersity uses a mixed approach that combines two of these techniques with interconnected views (Figure **??**). At the top are the two original trees allowing side-by-side comparison. Below them a third aggregated view called DiffTree shows the differences between the original trees. The three views are interconnected: selecting one node highlights and centers the two other corresponding nodes in the other views.

TreeVersity also displays the differences between the trees in a table representation (at the top left). The table lists all the nodes currently displayed, also with tightly coupled highlighting. At this time the columns includes the name of the node, number of children, level in the tree, and absolute and relative differences of each attributes. Other metrics could easily be added. Sorting columns allows the rapid selection of nodes with extreme values (e.g. largest relative difference or larger number of children).

Calculating the DiffTree

To build the DiffTree we compute the difference in all the attributes for each original nodes. For this we require the nodes to be uniquely identified by one of the attributes (typically the label of the node); with this identifier we can match the nodes and compute the differences of the identical attributes (numerical only for now). The set of nodes that do not have a match on the opposite tree are the topological differences.
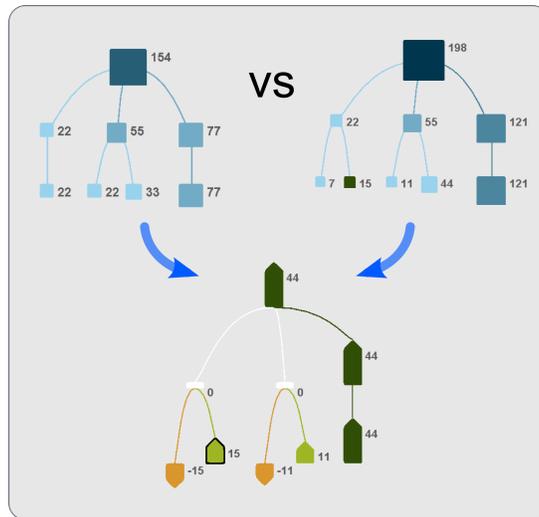
Figure 4: DiffTree construction. For each node a bullet is created representing the amount of change (absolute or relative) on it. The DiffTree can be calculated using all the nodes from the original tree or with a selection of those, like only with the nodes created or removed, or only the nodes that appear in both trees. In order to maintain the tree characteristics of the structure, a rule that guarantees that the ancestor of a visible node should be always visible is executed after any application of a filter by the Users.

We calculate the difference between the nodes' values of the left tree and the nodes' values of the right tree. A positive difference indicates that the value on the right is larger than on the left. A node present on the left but not on the right is considered a removed node and its value in the DiffTree will appear as negative, assuming the value of absent nodes as zero. This makes the placement of the original trees important. For example it is better (but also natural) to place 2011 on the left and 20012 on the right. Created nodes present in the right tree only will have a positive value. For each identification we create a node in the DiffTree that contains the values of the attribute differences in the corresponding nodes. Each node in the DiffTree is placed as a child of the node that corresponds to the original node's father.

By default the initial DiffTree shows the union of the original trees and contains all the nodes from both original trees. However users can filter out specific nodes by differential amounts and/or by topological characteristics (created, removed or present in both trees). The nodes are sorted according to the amount of change (absolute or relative). The DiffTree is sorted first, then the same order is applied to the original trees.

Tree visualization

Different tree visualizations were considered for both the original and DiffTree views, and after a process of selection, the node link representations were chosen. In particular

the treemap was eliminated because - while it shines at showing leaf node values - it cannot show values for internal nodes and does not show the topological structure clearly. The node-link representation seemed to be more versatile to address the four types of tree comparison we wanted to address.

Users can choose a left-to-right (horizontal) or top-to-bottom (vertical) layout. The original trees use rectangular icons with color and size redundantly encoding the attribute values of the nodes; the color and size scale uses the maximum possible values found in both trees combined so that the ranges in both original views are the same, facilitating side-by-side comparison. The DiffTree view uses a novel glyph visualization—the Bullet—to represent differences between the two original trees.

The Bullet for encoding change

The Bullet glyph encodes the cardinality of the change, the amount of change, and the creating/deletion. The shape's direction represents the cardinality of the change: left for negative and right for positive in the horizontal layout and down for negative and up for positive in the vertical layout. The bullet size represent the amount of change. Color is used to encode both the cardinality and amount of change in the nodes. To accommodate for color blindness users can select from preset color palettes that are binned in five steps to ease differentiation. The colors in the DiffTree are different from those in the original trees in purpose, because they usually need very different value ranges. White rectangles represent nodes where the amount of change is zero. Finally thick white or black borders are added around the bullet to denote removed or newly created nodes (new for added nodes, black for removed). By default both size and color are redundantly encoding the absolute amount of change (e.g. the amount in dollars in the case of a budget), but users can switch to relative change (i.e. percent change), or assign color and size to different characterization of the changes.

Interaction

Filtering

Users can filter the nodes by topological change, by range of values, and by maximum depth. Topological change allows users to see only the nodes that were created, or removed, or that are present on both trees. With the filter-by-node-variables range, users can keep visible only nodes whose value fall within a specific range, using an absolute or relative amounts of change. Finally, the filter by maximum depth hides all the nodes that are deeper than a specified maximum depth.

After the filtering operation, any nodes that do not fit the new selection criteria will be hidden from all the views (including the table), and then by an animation, the empty space will be reclaimed for the remaining nodes, making optimal use of screen viewing space.

Figure 5: The Bullet representation. Shapes going up (or right for horizontal layouts) represent nodes with increases in their values (those where the value on the tree of the right is bigger than the correspondent node value on the tree of the left) while decreases are represented with shapes going down (or left). The size of the shape represent the amount of absolute (or relative change) compare with the rest of the nodes of the tree, the biggest shape corresponds to the node with the maximum value overall and the rest are normalized according to it.

Overview

All three visualizations offer panning and zooming options for navigation. However when analyzing trees with thousands of nodes, a zoomed out (macro) view of the whole tree can produce a cluttered mass of nodes. For this TreeVersity offers an option that distributes the distance between the layers of nodes to fit the screen, as seen in Figure 7. This option is especially useful to understand the structure of the compared trees and of the DiffTree.

Navigation

Users can focus on a subtree comparison. This is done by double clicking on the root node of the subtree of interest, as seen in Figure 6. After navigating into a subtree all the views will be updated to display only the nodes on it; this is particularly useful in decluttering the screen. A navigation panel records currently navigated nodes and allows users to return to a previously navigated node. All of TreeVersity's navigation and filtering options are available for the newly navigated in subtree.

Labels and Colors

TreeVersity offers multiple options to control the visualizations. Users can display the node's values and other descriptive information as an adjacent label. Users have control of how much information is provided (name of the node, its value, relative values, and other descriptions). TreeVersity maximizes spatial considerations when displaying the
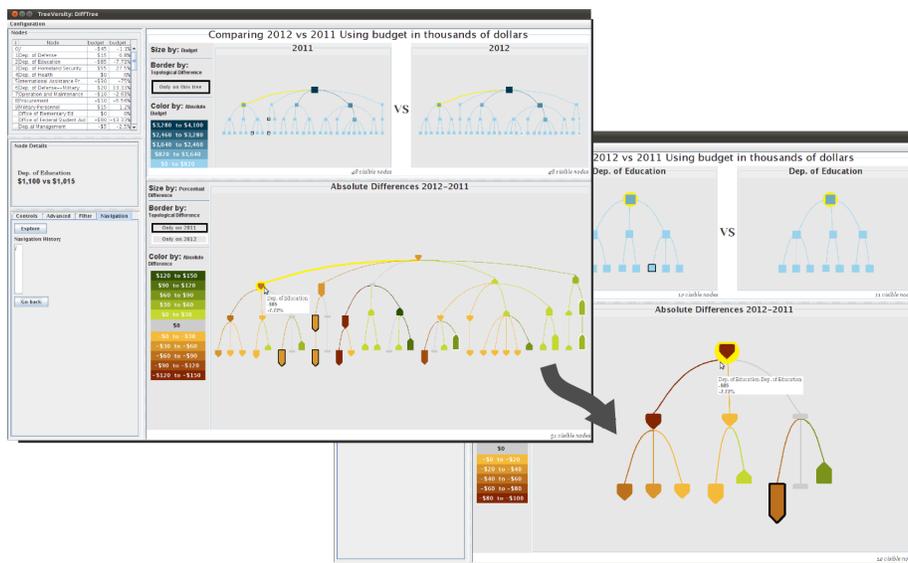
Figure 6: TreeVersity's Navigation function. After looking at the whole tree comparison (on the left), the user has decided to explore in more detail the "Department of Education", so he/she double clicks on the node, and TreeVersity recalculates the DiffTree for the subtree rooted on the selected node (on the right). After the navigation the colors and sizes are recalculated according to the new maximum values of the subtree.

layout of the nodes and their corresponding labels, and, if necessary, users can select the option to truncate the labels.

The colors and size of the Bullets on the visualization can represent either the absolute or relative values (and differences) of the nodes. By default the variable used for sorting the nodes is the same as the one used for coloring, so if users change the coloring the nodes will be rearranged on all views, using animation, to fit the new ordering scheme.

EXAMPLE OF USE

To illustrate TreeVersity's functionalities, we uses the 2011 and 2012 U.S. Federal Budget outlays [2] as published on the White House website [3]. The budget is a tree composed by a maximum of five levels classified as Agencies, Bureaus, Accounts, Sub-functions and the Type of Account (if it is Discretionary or Mandatory). All the values are in thousands of dollars. The budget for 2011 is represented by a tree of 4,103 nodes, and 2012 has 4,024 nodes.

Overview of the changes

Looking at the overview of the differences between the two years we could see that differences were very large and immediately switched to a log scale for the colors palette. To focus on the high level changes, the trees were filtered by a maximum depth of 2 showing only Agencies and Bureaus (Figure 7). With so many nodes the bullets are not readable but we can still see that the top bullet is red telling us that the budget for 2012 is smaller than in 2011. By looking at the color of the branches we can see that increases and decreases are spread fairly evenly over the entire government. We can also see that there is not a single Agency with more than one Bureau that maintains a consistent budget in 2012; all of the unchanged Agencies (colored gray) have only one Bureau. Users can use the zoom and filter options to reveal more details of these local changes.

Exploring addition and deletions

After looking at the overview we can look at the 273 created and removed nodes using the filters on the left panel. The U.S. Department of Transportation and U.S. Department of Treasury were the two Agencies with the biggest number of created and removed nodes with 32 and 27, respectively. By filtering the view to a maximum depth of 2, TreeVersity shows the only 9 creations and deletions at the top levels of the Budget (Figure 8). One of the addition is the "Allowances" Bureau with a budget of $26.67-billion. It was created in 2012 within the "Department of Defense", which is quite interesting given that, despite this increase, the department of Defense had an overall decrease of $67.88-billion.

---

[2]Amount of money that is expected to be spent

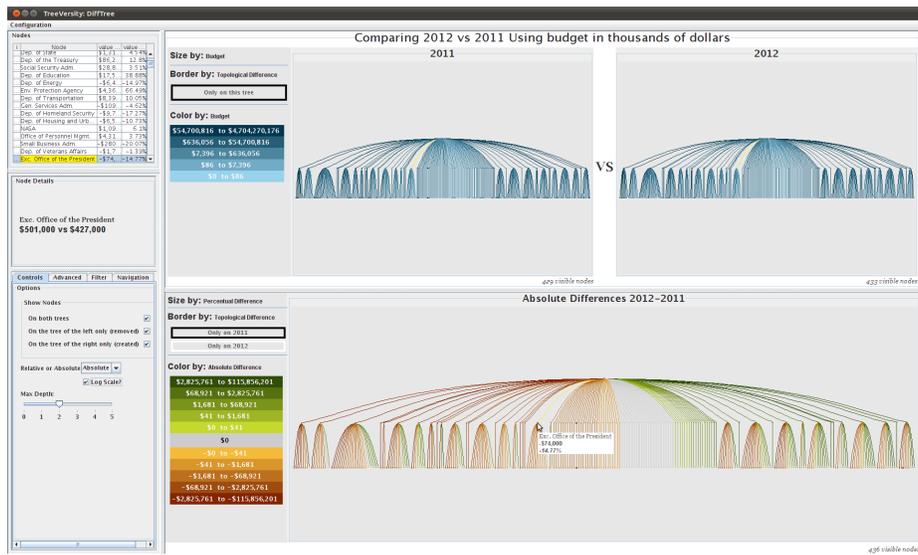[3]http://www.whitehouse.gov/omb/budget/Supplemental/

Figure 7: Overview of the first two levels of the Federal Budget Comparison, the Agencies and Bureaus, showing absolute change (in millions of dollars). We can easily spot the agencies whose budget did not change (light gray). By doing an spatial match with the original trees, is possible to see that those are Agencies with small budgets. We can see that are many Agencies with only one Bureau. Looking at all the agencies whose budget increased that have have more than one bureau we can see that for all of them there is stil a redistribution within the agency as there is a balanced number of Bureaus that increased and bureaus that decrease. On the other hand Agencies who lost money have all their Bureaus also reduced. Only one Agency ("Executive Office of the President") was reduced in every single one of its Bureaus and is entirely red.
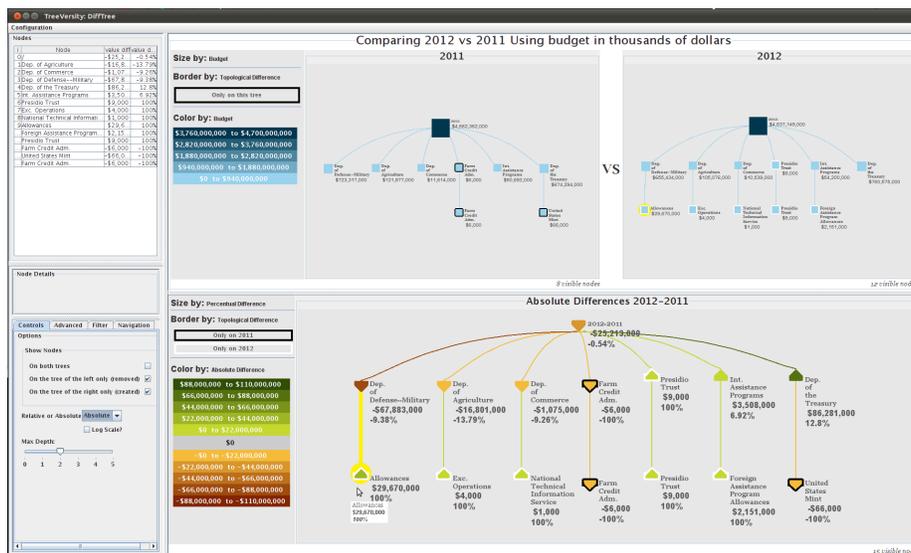
13

Figure 8: Created and Removed Agencies (nodes at level 1) and Bureaus (nodes at level 2) in the Federal Budget between 2012 and 2011. Only one Agency was removed "Farm Credit Administration" (with a budget of $6,000,000) and one was created "Presidio Trust" (with $9,000.000 of budget). At the Bureau level 5 nodes were created and 2 removed. On the top part of the screen, TreeVersity displays the correspondent created and removed nodes in the original trees, in their respective hierarchies. To preserve the hierarchy the ancestors of the created or removed nodes are also drawn.
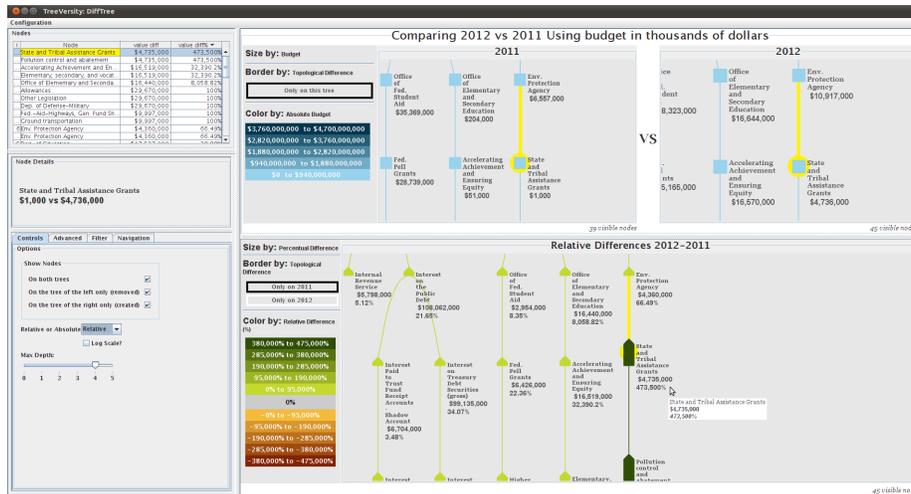
Figure 9: By ordering the table by relative change users can find the nodes with largest relative increase or decrease. Clicking on a row centers the display on the corresponding node. Here the largest relative increase in for "Pollution control and abatement". The node goes from $US1,000,000 in 2011 to US$4,736,000,000 in 2012, a relative increase of 473% which would be hard to find by browsing this very large tree.

Finding accounts with big relative changes

To find the biggest changes anywhere in the budget the tabular view can be useful. Sorting by the amount of relative change we can see in the top rows where the biggest changes occurred. By selecting a row in the table, TreeVersity highlights and pans the views to center on the correspondent node. The DiffTree was then configured to show the relative change of each node between the two years. Here, the Sub-function "Pollution control and abatement" and its Account "State and Tribal Assistance Grants" rises to the top of the table with an increase of 473% (their budget went from $1-million dollars to $4.73-billion) (Figure 9, see the big dark-green bullets). This is an outstanding difference compared to the rest of accounts, even with the budget of its parent, the Environmental Protection Agency.

USER STUDY

We conducted a user study with 8 participants to evaluate if users could understand the visual encodings and the basic interface organization of Treeversity without any training at all. The dataset was presented as the budget of an hypothetical country but was in fact a small subset of the USA Federal Budget for 2011 and 2012, with values modified to include multiple interesting sets of changes that could be easily spotted by the users (if they could interpret the encodings accurately), e.g. high increases, accounts receiving budget increases in a department where all other accounts where getting cuts;
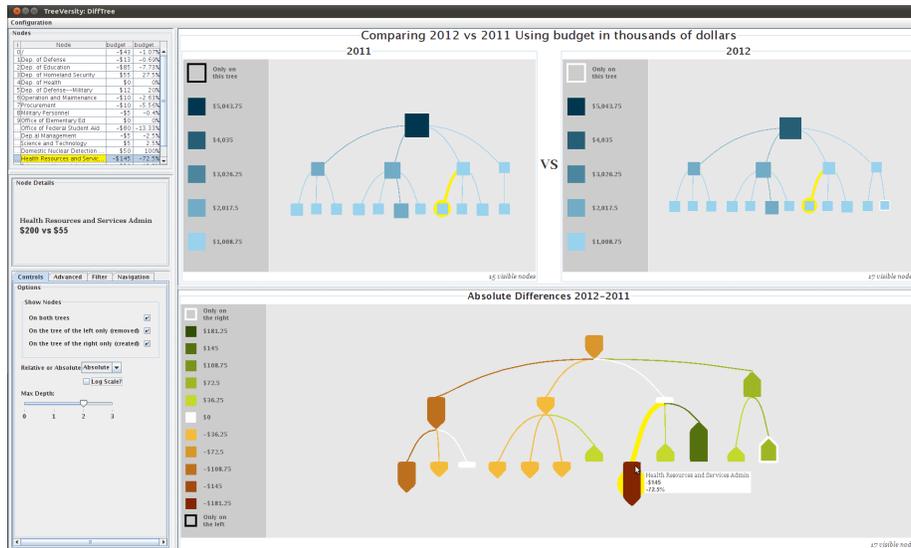
Figure 10: Screen shot of the TreeVersity's version used in the user study. Note the older legend.

accounts created and removed at different levels. The dataset had 46 nodes distributed in four levels with 1 node removed from 2011 and 3 created on 2012. Participants were new students in their first week of the Master on Human Computer Interaction program, three females and five males. Their background varied from Computer Science to Design and Mathematic. They had never heard of TreeVersity. After a short one minute introduction explaining the objective of TreeVersity and the nature of the dataset, participants were asked to try the interface and to describe their perception and understanding of the interface using a think aloud protocol. No training was provided at all. At the start the tree was opened at level 3 so showing only 17 nodes (as shown in Figure 10). Note in the figure that we used a slightly older version of TreeVersity than described earlier in the paper. The main difference was the design of the legend, and while was used for both new nodes and nodes with a value of zero which caused confusion. .All the desktop interactions and discussions were recorded.

For about five minutes participants explored on their own while the observer kept track of what had been correctly interpreted and learned (or not), using a checklist of expected concepts to be discovered. After five minutes misunderstandings were discussed and participants questions answered. If a particular feature had not been mentioned in the think aloud exploration, the observer pointed out the feature and asked participants to comment on what it might represent or do.

Overall we were pleased to see that without any training all participants grasped and correctly interpreted most of interface components of TreeVersity. The first thing participants described was always the side-by-side trees and the DiffTree. They all correctly described the relationship between the three trees, and even if the tight coupling of the highlighting between the views wasn't always described explicitly, all of the

users started using it right away. Then participants talked about how they interpreted the glyphs. For the side-by-side trees, everyone immediately associated color with the amount of money at each node, however some people overlooked the size property (both color and size encoded the same information).

While the DiffTree is much more complex we were pleasantly surprised by how well participants could learn to interpret the display on their own. By looking at the matching nodes on each view, and the shape and color of the bullets, people were able to conclude that each node on the DiffTree was representing the amount of change on that node. The direction of the change was always understood correctly, by commenting on the color and direction of the bullet. Some people focused solely on the color of the bullet and seemed to ignore the meaning direction and size, while others guided themselves by the shape alone and seemed to ignore the color. Since size and color encoded the same information it was appropriate. Several people had problems understanding the older color legend used in the study (Figure 10). They only saw one number on the scale and did not guess that each color represented an interval. However other participants thought it was obvious, for example saying "I can't see why somebody will interpret this in a different way". Participants had no problem understanding that nodes being small and white meant that their value had not changed, even in situations where a internal node had not changed but all of its children had changed significantly (with the sum of the changes being zero).

In the user study setup size and color encoded the same information, but the meaning of the size of the bullets was not mentioned in the legend. To our surprise four subjects assumed that the size was representing the percentage change while the color was representing the absolute change. This was wrong guess of the meaning of the visualization, but we hope that including size coding at the top of the legend will fix this misinterpretation. This suggests also that encoding both variables at the same time might in fact be a good idea as a default encoding as it fits the expectation of some users.

The topological differences (represented with a black or white thick border), where usually unnoticed initially, but all users eventually noticed them. They didn't seem to immediately understand what they meant, but figured it out either by looking at the legend or by using the coordinated views and noticing the node was missing in one of the views. Some users suggested that labels in the legend could be more meaningful e.g. "only in 2011" instead of "on the left only". The black and white colors we used initially to denote the topological changes were found confusing because white was already associated with nodes without change. We later changed the coding of zero as gray intead of white.

After the initial free exploration the participants were shown the larger tree with 46 nodes and asked to find the most significant changes. All of the subjects easily found many insights in the data. They followed a pretty consistent process: they started by looking at the created and removed nodes, then pointed out the nodes with the biggest differences, both negative and positive. Then most subjects took a step back to describe the large overall negative or positive changes in the budget between the two years. Finally some of the participants pointed out the more subtle patterns (e.g. a node getting a big increases while all the sibling nodes are being cut). Those who did not

spontaneously find those patterns they were able to find them when asked to look for them.

We were curious how people might understand relative changes so we asked participants to explain how they thought the relative percent change was calculated. While they all had been able to read and interpret the absolute and relative differences correctly, more than half of the participants struggled to explain correctly how it was calculated which confirms how complex the task of comparison can be. Finally they were given the opportunity to review the operation and labeling of the controls and suggest improvements.

After the test participants were asked three questions about the usefulness of TreeVersity on a scale of 1 through 7, were 1 was "not useful" and 7 "very useful". All answers were above 5, and six out of the eight participants answered the three questions with 6s and 7s.

CONCLUSIONS

We introduced TreeVersity, a new interactive visualization that gives users powerful tools to detect both node value changes and topological differences. We believe that our new approach allows users to easily identify differences between trees, see patterns and spot exceptions. A usability test confirmed that - even without training - users could use the interface and understand the visual encodings embedded in the bullet glyph. While we are pleased with the early results the problem of comparing trees is very challenging when there are many nodes and levels, and a large number of created and deleted nodes. Dealing with labels and with skewed distributions of absolute and relative distribution changes will also require special attention. Additional user studies will be needed to guide TreeVersity designs regarding how much information can be displayed at a time, for example it is not clear that all users can comfortably read glyphs that show two separate variables with size and color.

Current work includes working with transportation data analysts to compare incident management statistics between state transportation agencies, as well as studying the evolution of topics in transportation publications over the years. We are also started developing alternate visualizations to the node-link diagrams, such as a treemaps and icicle trees which might be appropriate for trees with aggregated values in internal nodes and no topological changes. Future work includes comparing collections of trees and generating reports that guide users through the most important differences between the trees.

ACKNOWLEDGEMENTS

References

[1] Amenta, N., and Klingner, J. Case study: Visualizing sets of evolutionary trees. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on* (2002), 71–74.

[2] Auber, D., Delest, M., Domenger, J. P., Ferraro, P., and Strandh, R. EVAT: environment for vizualisation and analysis of trees. *IEEE InfoVis Poster Compendium* (2003), 124–125.

[3] Battista, G. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry 4*, 5 (Oct. 1994), 235–282.

[4] Bille, P. A survey on tree edit distance and related problems. *Theoretical Computer Science 337*, 1-3 (June 2005), 217–239.

[5] Bremm, S., von Landesberger, T., and Hamacher, K. Interactive visual comparison of multiple phylogenetic trees. http://www.gris.tu-darmstadt.de/research/vissearch/projects/ViPhy/, Oct. 2011.

[6] Card, S. K., Suh, B., Pendleton, B. A., Heer, J., and Bodnar, J. W. Timetree: exploring time changing hierarchies. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, vol. 7, IEEE (2006), 3–10.

[7] Fisher, D., Dhamija, R., and Hearst, M. Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization*, vol. 2001, IEEE (2001), 43 – 50.

[8] Furnas, G. W., and Zacks, J. Multitrees: enriching and reusing hierarchical structure. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, ACM (New York, NY, USA, 1994), 330–336. ACM ID: 191778.

[9] Ghoniem, M., and Fekete, J. D. Animating treemaps. In *Proc. of 18th HCIL Symposium-Workshop on Treemap Implementations and Applications* (2001).

[10] Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., and Roberts, J. C. Visual comparison for information visualization. *Information Visualization* (2011).

[11] Graham, M., and Kennedy, J. Combining linking and focusing techniques for a multiple hierarchy visualisation. In *Information Visualisation, 2001. Proceedings. Fifth International Conference on* (2001), 425–432.

[12] Graham, M., and Kennedy, J. Exploring multiple trees through DAG representations. *IEEE Transactions on Visualization and Computer Graphics* (2007), 1294–1301.

[13] Graham, M., and Kennedy, J. A survey of multiple tree visualisation. *Information Visualization* (2009).

[14] HCIL. Infovis benchmark - PairWise comparison of trees. http://www.cs.umd.edu/hcil/InfovisRepository/contest-2003/, Aug. 2011.

[15] Herman, I., Melançon, G., and Marshall, M. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on 6*, 1 (2000), 24–43.

[16] Hong, J. Y., D'Andries, J., Richman, M., and Westfall, M. Zoomology: comparing two large hierarchical trees. *Poster Compendium of IEEE Information Visualization* (2003).

[17] Johnson, B., and Shneiderman, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization (Vis)*, IEEE (1991), 284 – 291.

[18] J\ürgensmann, S., and Schulz, H. Poster: a visual survey of tree visualization. *Proceedings of IEEE Information Visualization (Salt Lake City, USA, 2010), IEEE Press*.

[19] Kruskal, J. B., and Landwehr, J. M. Icicle plots: Better displays for hierarchical clustering. *The American Statistician 37*, 2 (1983), 162 – 168.

[20] Lee, B., Robertson, G. G., Czerwinski, M., and Parr, C. S. CandidTree: visualizing structural uncertainty in similar hierarchies. *Information Visualization 6*, 3 (2007), 233–246.

[21] Margush, T., and McMorris, F. R. Consensusn-trees. *Bulletin of Mathematical Biology 43*, 2 (Mar. 1981), 239–244.

[22] Mohammadi-Aragh, M. J., and Jankun-Kelly, T. J. MoireTrees: visualization and interaction for multi-hierarchical data.

[23] Morse, D. R., Ytow, N., Roberts, D. M., and Sato, A. Comparison of multiple taxonomic hierarchies using TaxoNote. In *Compendium of Symposium on Information Visualization* (2003), 126–127.

[24] Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., and Zhou, Y. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In *ACM SIGGRAPH 2003 Papers*, ACM (San Diego, California, 2003), 453–462.

[25] Parr, C. S., Lee, B., Campbell, D., and Bederson, B. B. Visualizations for taxonomic and phylogenetic trees. *Bioinformatics 20*, 17 (2004), 2997.

[26] Plaisant, C., Grosjean, J., and Bederson, B. B. SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proceedings of the IEEE Symposium on Information Visualization*, IEEE (1998), 57–64.

[27] Schulz, H., Hadlak, S., and Schumann, H. Point-based tree representation: A new approach for large hierarchies. In *Proceedings of the IEEE Pacific Visualization Symposium*, IEEE (Apr. 2009), 81–88.

[28] Spenke, M. Visualization and interactive analysis of blood parameters with Info-Zoom. *Artificial Intelligence in Medicine 22*, 2 (May 2001), 159–172.

[29] Stockham, C., Wang, L., and Warnow, T. Statistically based postprocessing of phylogenetic analysis by clustering. *Bioinformatics 18*, Suppl 1 (July 2002), S285–S293.

[30] Thorley, J. L., and Page, R. D. RadCon: phylogenetic tree comparison and consensus. *Bioinformatics 16*, 5 (2000), 486.

[31] Tu, Y., and Shen, H. Visualizing changes of hierarchical data using treemaps. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1286–1293.

[32] Wattenberg, M. Visualizing the stock market. In *CHI'99 extended abstracts on Human factors in computing systems* (1999), 188–189.