Visualisation of hierarchical multivariate data: Categorisation and case study on hate speech

Article *in* Information Visualization · September 2022

DOI: 10.1177/14738716221120509

8 authors, including:

Ecem Kavaz
University of Barcelona
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Anna Puig
University of Barcelona
82 PUBLICATIONS **427** CITATIONS
SEE PROFILE

Inmaculada Rodriguez Santiago
University of Barcelona
62 PUBLICATIONS **518** CITATIONS
SEE PROFILE

Montserrat Nofre
University of Barcelona
7 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Info Vis

Article

Information Visualization
1–21

DOI: 10.1177/14738716221120509
journals.sagepub.com/home/ivi
SAGE

Visualisation of hierarchical multivariate data: Categorisation and case study on hate speech

Ecem Kavaz[1], Anna Puig[1,2], Inmaculada Rodríguez[1,3], Reyes Chacón[1], David De-La-Paz[1], Adrià Torralba[1], Montserrat Nofre[1,4] and Mariona Taule[1,4]

Abstract

Multivariate hierarchical data has an important role in many applications. To find the best visualisation that best fits a concrete data is crucial to explore and understand the relationships between the data. This paper proposes a categorisation – Elongated and Compact – of hierarchical data based on the inner shapes of the hierarchies, that is the connectivity degree of the internal nodes, the number of nodes, etc, that can be applied to any hierarchical data. Based on this taxonomy, we explore implicit and explicit layouts – Tree, Circle Packing, Force and Radial – to provide users with a complete view of the data. We hypothesise that Tree and Circle Packing fit with Elongated structures, and Force and Radial fit with Compact ones. In addition, we cluster multivariate features to embed them in the hierarchical layouts. Especially, we propose two different glyphs – *one-by-one* and *all-in-one*, and we bet for the *one-by-one* glyphs as the most suitable for showing the distribution of several features along with the hierarchical structures. To validate our hypotheses, we conducted a user study with 35 participants using a hate speech annotated corpus. This corpus comes from 4359 comments posted in online Spanish newspapers. The results indicated that users preferred the Tree layout over the other three layouts (Circle, Force, Radial) with both types of structures (EC and CC). However, when we focused the analysis only on Radial and Force layouts, both of them scored significantly higher with Compact than with Elongated data. Moreover, participants scored the *one-by-one* glyph higher than the *all-in-one* glyph, but the difference was not significant.

Keywords

Hierarchical visualisations, multivariate data, hate speech analysis

Introduction

Nowadays the massive amount of data generated by social networks, digital media, and society in general, is difficult to track and analyse not only because of its sheer volume but also because of its complex interrelationships. Most of these data have hierarchical relationships, meaning they are related to each other in parent-child relationships.[1] Indeed, many research areas need to analyse hierarchical data, such as taxonomy of language terms in linguistics,[2] organisational structures in business,[3] genomics in biology,[4] and related comments in social media.[5]

[1]Departament de Matemàtiques i Informàtica, Universitat de Barcelona (UB), Barcelona, Spain
[2]Institut of Complex Systems (UBICS), Universitat de Barcelona (UB), Spain
[3]Institut de Matemàtica UB (IMUB), Universitat de Barcelona (UB), Spain
[4]Facultat de Filologia i Comunicació, Universitat de Barcelona (UB), Spain

Corresponding author:
E Kavaz, Departament de Matemàtiques i Informàtica (UB), Universitat de Barcelona (UB), Av. Corts Catalanes, 585, Barcelona 08024, Spain.
Email: ekavazka27@alumnes.ub.edu

Generally, these hierarchies are so complex, big, and multivariate that dealing with such intricate structures is a challenging task. Therefore, visualisations allow analysis of complex hierarchical data, as they can present a variety of information as well as help people convey complex information more effectively and quickly.[6,7] Indeed, there exist a large number of well-known visualisation methods to show hierarchies (e.g. Treemap, Tree diagram, Sunburst).[8]

However, the main dilemma is in really finding the visualisation that best fits a concrete hierarchical structure. In fact, hierarchical structures have different *shapes* depending on the connectivity degree of the internal nodes, the number of nodes, etc. By shapes, we mean that hierarchies present different nodes' distributions, that is with non-fixed levels and non-fixed fan-outs (broader versus narrow and deeper). Some works[9] propose hierarchical visualisations taxonomies based on dimensionality – that is 2D or 3D –, and on nodes' alignment, but it remains the challenge of identifying the best well-fitted method depending on the inner shape of the data, before placing the data on a canvas.

Moreover, we emphasise the 'Overview first' of Shneiderman's information-seeking mantra[10] "Overview first, Zoom and Filtre, then Detail on Demand". That is, a complete view of a hierarchical structure is the very first step to understand and interpret data, followed by partial views that allow the analysis of specific details. Moreover, overviews are appropriate for comparison of multiple hierarchical structures, that is multiple data sets. For example, a hierarchical visualisation of replies to comments of an online news article can be compared to others to extract first sight knowledge such as the news that generated more toxic comments.

Furthermore, when the data is also multivariate (i.e. each data point is characterised with a number of features), solely hierarchical visualisations are not sufficient to communicate high numbers of features at once. Therefore, the use of visual variables like colour, position and shape, along with icons and glyphs can help to analyse and communicate more information in a better way.[11,12]

Based on the assumption of an existing variety of hierarchical structures characterised by their shape, this paper formalises their categorisation in *Elongated (narrow)* and *Compact (broad)* structures and argues for the adequacy of a visualisation method – Tree, Circle Packing, Force and Radial (see Figure 1) – depending on defined attributes, such as the growing factor and the number of direct children of a node, which can be applied to any hierarchical dataset. This paper also contributes with a formalisation of features of multivariate data and, consequently, integrate their visualisation in a hierarchical structure. Based on this formalisation, we introduce two types of glyphs: (i) the *one-by-one*, where features are depicted by coloured dots placed one next to each other and (ii) the *all-in-one*, where a single pie chart represents all the features. Thus, we argue that a *one-by-one* glyph is more informative than a *all-in-one* glyph for depicting multivariate data in overviews of hierarchical structures.

In summary, we aim to achieve a global overview of the whole hierarchical structure where both parent-child relationships and features' distribution can be clearly analysed without overwhelming the user perception. Thus, our hypotheses are defined as follows:

- ***H1***: *When the hierarchy is categorised as Elongated (EC), the most informative methods to visualise it are Tree layout and Circle Packing* (see layouts (a) and (b) in Figure 1).
- ***H2***: *When the hierarchy is categorised as Compact (CC), the most informative methods to visualise it are Force layout and Radial layout (see layouts (c) and (d) in* Figure 1*).*

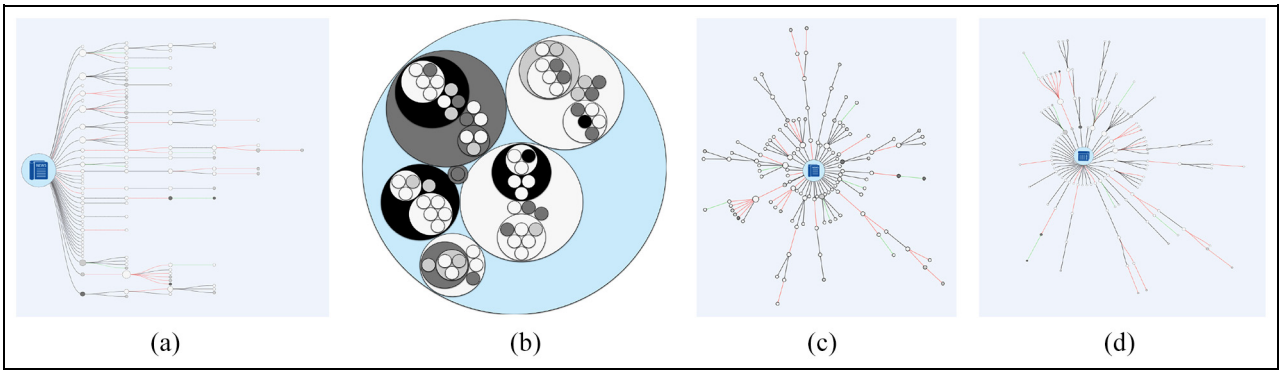


Figure 1. Four hierarchical layouts considered in this research: (a) tree layout, (b) circle packing, (c) force layout and (d) radial layout.

- ***H3*: *With a large number of features, the most informative glyph to embed them in a hierarchical layout is the one-by-one glyph instead of an all-in-one glyph.***

We tested these hypotheses using an in-house tool http://datavisualizationinlinguistics.herokuapp.com/ created for the analysis of social data. Concretely, our case study was the NewsCom-TOX corpus that was developed to analyse hate speech contained in the online news. The data were collected from 21 online news papers' comment sections and annotated with 14 features to detect the toxicity of the comments (constructiveness, sarcasm, level of toxicity, etc.) We used hierarchical multivariate visualisations to present this corpus and conducted an evaluation with 35 participants that scored the four layouts in tasks designed to validate our hypotheses.

The results indicated that we can partially accept H1 and H2 since users preferred the Tree layout over the other three layouts (Circle, Force, Radial) with both types of structures (EC and CC). However, when we focused the analysis only on Radial and Force layouts, both of them scored significantly higher with CC than with EC, providing additional support to H2. Moreover, related to H3, *one-by-one* glyph scored higher than *all-in-one* but without significant differences.

Related work

Visualisation of hierarchical data

There exist some previous works that explore hierarchical visualisation techniques. They are mainly classified into two categories: implicit and explicit.[13] Implicit hierarchical visualisations represent parent-child relationships with positional encoding using shapes within other shapes (e.g. using rectangles both in treemaps and icicle diagrams, and circles in circle packing), while on the other hand explicit visualisations represent these relationships with lines (e.g. tree layout). Another proposal presented four types of layouts: pack layouts (circle packing), node-link layouts (tree and radial layout), partition layouts (sunburst diagram and icicle diagram) and treemaps,[14] explaining when to use each visualisation depending on the type of the data or the task to perform. For instance, they recommended using partition layouts for analysing numerical data, and node-link layouts for analysing paths. Our research also believes in studying suitable layouts but relying on the inner shape of the hierarchical data.

Several researches explored visualisations of hierarchical structures in various applications. For instance, Darzi et al.[15] designed a radial layout to visualise omics data (e.g. genome, proteome). They selected radial layout as they presumed that tree layout was insufficient with a large number of nodes and scaled poorly. Nevertheless, we consider the problem relays on the distribution of the nodes rather than the number of nodes, as radial layout may look shapeless, and thus, less comprehensible for the user when the data is narrow and slender, which we aim to test experimentally in this research.

Moreover, GrouseFlocks[16] focused on taking an input hierarchy and showing other related hierarchies of it (e.g. from all movies to action movies) using several layouts, they specifically used both tree and circle packing. While we find interesting to explore hierarchies inside others using several layouts, circle packing gets crowded and begins to visualise hard-to-follow nesting when the data structure gets deeper and wider. In this research, we consider other explicit layouts for deeper and wider hierarchical structures such as radial and force-based.

The literature also provides several works closely related to our case study of hate speech analysis in conversations. For example, ConVis[17] was designed to analyse comments in online conversation threads and focused on getting the perception of the whole conversation at first glance. They displayed each comment as a horizontal stack bar and all the replies are stack bars placed under each other by their order in the thread. The levels of the threads are shown by positioning bars with indentation. However, this tool might be not sufficient to visualise a complete view of deeper or broader hierarchies due to its architecture as these kinds of hierarchies might not fit on to the screen. Moreover, as it uses indentation to show the levels of threads, if the data sets have several long and narrow threads, the bars of these threads will be stretched to the right and they will leave a lot of empty space between these long spines. Another example is ShareFlow[18] which used a radial tree layout to show information diffusion between individuals on social media fan pages' comment sections. Nevertheless, their method did not show the data hierarchically and rather showed hierarchical items side by side.

Additionally, Forum Explorer[19] was designed to visualise threaded conversations on websites like Reddit. They used a radial layout to visualise the threads as a whole, that is achieve a complete view of all the threads, and, similarly as[12] did, they used a traditional tree layout additionally to the radial layout to visualise some of the large sub-conversations in the hierarchies separately. This technique can be useful to analyse larger sub-threads in detail however, we focus on visualising conversations as a whole with appropriate layouts depending on the shape of hierarchical data, and radial layout might not be the most informative layout for all kinds of hierarchical structures.

Furthermore, VizWick[20] is a web-based tool that was designed to provide visualisations for hierarchical data. While they emphasised that only one visualisation layout is not enough to visualise all hierarchies as they have different properties such as size, depth, and branching factor, they tried to solve this problem by introducing a multiple views dashboard. Authors suggested that visualising a single dataset simultaneously with different layouts can give more analytical information about it. They included five visualisation methods, circle packing and sunburst between others, and up to four windows to visualise the dataset. This approach can be useful as each visualisation can offer information about a different viewpoint of the data. However, we detect the most informative viewpoint of a dataset, by categorising its inner data distribution.

Visualisation of multivariate data

Creating an understandable multivariate visualisation is a difficult job as these kinds of hierarchies are very complex due to the variety of the information that can be stored in them.[12] The main goal is to support visualising more than a few attributes (that are usually limited to using colour, shapes and size) while not losing comprehension of the hierarchies.

The literature shows works on multivariate data visualisation following different approaches and contexts. These works proposed different techniques for visualising multivariate data such as glyph-based,[21] interaction-based, icon-based,[22] hybrid visualisations[23] and animation-based.[24] For example,[25] used the method of reducing the number of graph elements shown on the active view by displaying multivariate data in pairs one at a time. However, it can be necessary to analyse more than two attributes in some cases. Also, this method might cause information loss while navigating through different views. While[18,26] used various filtres and directly mapped attributes (e.g. colour) onto the main visualisation (e.g. node-link graph) to communicate multivariate data,[27] used an icon-based approach. These approaches have high potential since if they were combined together, they would communicate more information, provide visualisation options for different kinds of multivariate data and some of the approaches can be used on-demand to reduce the clutter on the visualisation.

Moreover, glyph-based visualisation is a popular option. A glyph is a small visual object that represents several attributes.[28] It can be used individually[29] and also in combination with other graphs[30] to add more meaning to the data being presented. For example, Social Wave[31] used glyph-based approach in cooperation with their main visualisation to analyse the distribution of popular hashtags (collected from Twitter) in various locations. Their main graph is a network graph but they created three glyphs to communicate more information about hashtags (e.g. proportion of used hashtags) and assigned these glyphs to the nodes according to their sizes. This is an interesting approach because glyphs added more information to their main visualisation as well as reduced the clutter by having different versions according to the size.

In the context of conversation analysis, ContoVi,[32] was designed to explore speaker behaviours in multiparty conversations. The tool has four main animation-based visualisations. In addition to these four views, they included an argumentation glyph to detail the features of each utterance. Ten argumentation attributes such as assurance, common ground, etc… were mapped onto an *all-in-one* glyph (similar to a pie chart) to explore the degree of justification and stances of the speakers in utterances. The main drawback of this approach is that the glyph is only shown in a separate view, requiring thus a change in the context of users' attention which can require retaining some details of the hierarchical structure and so alter the ability of the user to effectively process the information.[33] To conclude, the visualisation of a high amount of attributes requires the combination of several approaches such as icons and glyphs as well as visual variables like colour, shape and position.[34] According to previous works, we use several approaches to visualise multivariate data and, most importantly, we explore two strategies for using glyphs (*one-by-one* vs *all-in-one*) when needed for an overview visualisation of hierarchical structures.

Data categorisation

Hierarchical data categorisation

In this section we formalise hierarchical structures, introducing some basic definitions about hierarchies and their properties, which allow us to categorise their shapes as elongated and compact structures. This categorisation will later influence the display layout to be used for their visualisation.

We define a set of hierarchical multivariate structures,

$$\mathcal{T} = \{T^1, T^2, T^3, \ldots, T^n\} \quad (1)$$

where each T^k, the *k-th* hierarchy, is a directed rooted tree,

$$T^k = < Nodes^k, Edges^k >, \quad (2)$$

being,

$Nodes^k = \{n_0^k\} \cup N^k$ is the set of nodes of the hierarchy, being n_0^k the root node of T^k, and $N^k = \bigcup\{n_j^k\}$, $\forall 1 \leqslant j \leqslant n$, the set of all the other nodes of the tree T^k and $Edges^k = \bigcup\{e_{i,j}^k\}$, is the set of edges of the hierarchy, where $e_{i,j}^k$ is the directed edge from n_i^k to n_j^k, if n_j^k is related to n_i^k and n_i^k, $n_j^k \in Nodes^k$.

Notice that the size of T^k is the number of nodes of the tree, $\#Nodes^k = n + 1$, where n is the number of nodes directly or indirectly related to the root node, n_0^k. Moreover, all the hierarchies T^k are weakly connected and acyclic graphs. Each T^k is weakly connected since when we change all of its directed edges for non-directed edges, we get a connected non-directed tree, where there is one and only one path from any node to any other node in T^k. In addition, T^k, as a single rooted graph, it is acyclic, that is no node has more than one parent; and thus, it presents no cycles. We define a subtree of T^k rooted at n_j^k as $Subtree(T^k, n_j^k)$.

In our case study, related to online news articles and their comment sections, $\mathcal{T}$ are the set of news articles with their corresponding comments. One news and all its associated comments form a rooted tree, T^k, where the news article is the root node, n_0^k. Additionally, some users reply to n_0^k, and others reply to other comments, n_i^k, then edges symbolise all these direct replies.

Furthermore, there are some relevant properties of the hierarchy to bear in mind such as depth and width. First, let's consider the depth of any node of the tree T^k as the distance of the node n_j^k to the root n_0^k, taking as distance between two nodes (n_i^k, n_j^k), the number of connected edges from the node n_i^k to the node n_j^k,

$$depth(T^k, n_j^k) = distance(n_0^k, n_j^k), \forall 0 \leqslant j \leqslant n$$

Note that we can define similarly the depth of any node, n_i^k, of a subtree rooted in n_j^k:

$$depth(Subtree(T^k, n_j^k), n_i^k) = distance(n_j^k, n_i^k), \quad \forall n_i^k \in Subtree(T^k, n_j^k)$$

Then, we define $\mathcal{D}(T^k) = \max_{n_j^k \in N^k} depth(T^k, n_j^k)$ as the depth of the tree. Likewise the depth of a subtree is $\mathcal{D}(Subtree(T^k, n_j^k))$.

Secondly, we define $directChildren(n_i^k)$ as the set of nodes n_j^k belonging to T^k such that $distance(n_i^k, n_j^k) = 1$, and then we define $witdh(n_i^k)$ of a node as the number of its direct children,

$$width(n_j^k) = \#directChildren(n_j^k) \;\; \forall 0 \leqslant j \leqslant n$$

Thus, we define $\mathcal{W}(T^k) = \max_{n_j^k \in N^k} width(n_j^k)$ as the width of the tree.

Based on these characteristics, we will define how a tree grows. First of all, we define a node as significant when it has enough descendants in relation to its parent's descendants. That is, significant nodes fulfil the following condition:

$$\frac{size(Subtree(T^k, n_j^k))}{size(Subtree(T^k, parent(n_j^k)))} > tolerance \tag{3}$$

We define $significant(n_i^k)$ as the set of significant direct children of the node n_i^k.

Then, the Growing Factor of a tree rooted in n_i^k, either for the whole tree (T^k rooted in n_0^k) or any subtree ($Subtree(T^k, n_j^k)$ rooted in n_j^k), refers to how n_i^k branch out, that is it defines the relationship between its width and the width of any of its s-th subsequent levels. Within a level, we only consider those significant nodes, n_j^k. Thus, we define the Growing Factor of the subtree rooted in the node n_i^k of a s-th sublevel as:

$$GrowingFactor(n_i^k, s) = \frac{\sum_{n_j^k \in s-\text{th } level(n_i^k)} width(n_j^k)}{width(n_i^k)}$$

being $s - \text{th } level(n_i^k) = \{n_t^k\}$, where each node n_t^k is located at the s level of the subtree rooted at n_i^k, that is $depth(Subtree(T^k, n_i^k), n_t^k)$ equals to s.

Given any hierarchical structure – a whole tree or a subtree –, we define the tendency of its inner shape based on the Growing Factor. This trend tell us how the tree grows through its sublevels: elongated or compacted (see Equations (4) and (5), respectively). These conditions depend on a certain threshold values, N, L, $GF_{Elongated}$ and $GF_{Compact}$ related to the number of direct children, a certain number of levels and the growing factor thresholds related to elongated and compact tendencies, respectively. Next, we formalise the Elongated and Compact tendencies as a combination of two conditions AND ($\wedge$), and OR ($\vee$):

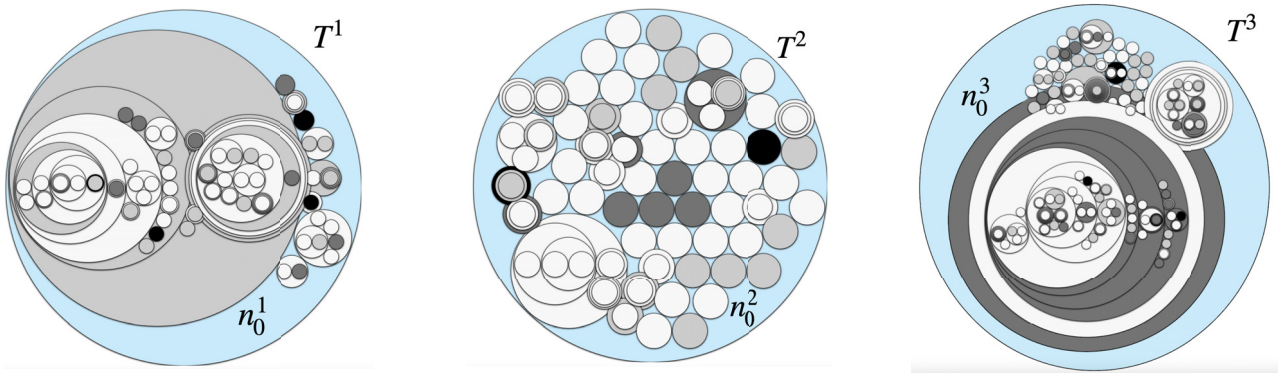
- Elongated Tendency, $ET(n_i^k, L, GF_{elongated})$ means that the tree rooted at n_i^k is narrow along l levels.

$$\begin{aligned} &width(n_i^k) \leqslant N \quad \wedge \\ &\big(\#significant(n_i^k) = 0 \quad \vee \\ &\quad GrowingFactor(n_i^k, s) \leqslant GF_{elongated}, \\ &\quad \forall s : 1 \leqslant s \leqslant L\big) \end{aligned} \tag{4}$$

- Compact Tendency $CT(n_i^k, L, GF_{compact})$ means that the tree rooted at n_i^k is broad along L levels.

$$\begin{aligned} &width(n_i^k) \geqslant N \quad \wedge \\ &\big(\#significant(n_i^k) = 0 \quad \vee \\ &\quad GrowingFactor(n_i^k, s) \geqslant GF_{compact}, \\ &\quad \forall s : 1 \leqslant s \leqslant L,\ GrowingFactor(n_i^k, s) \neq 0\big) \end{aligned} \tag{5}$$

Table 1. Circle packing with categories: (a) Elongated, (b) Compact and (c) Unspecified. The threshold values used to compute significant nodes and Growing Factors are: $N = 15$, $L = 4$, $GF_{Elongated} = 1.9$, $GF_{Compact} = 0.25$ and *tolerance* = 0.05.



(a) Data set categorised as Elongated	(b) Data set categorised as Compact	(c) Unspecified data set
$\mathcal{W}(T^1) = 18$	$\mathcal{W}(T^2) = 60$	$\mathcal{W}(T^3) = 45$
$\mathcal{D}(T^1) = 11$	$\mathcal{D}(T^2) = 4$	$\mathcal{D}(T^3) = 15$
$\#Nodes^1 = 100$	$\#Nodes^2 = 100$	$\#Nodes^3 = 200$
$width(n_0^1) = 9$	$width(n_0^2) = 60$	$width(n_0^3) = 45$
$\#significant(n_0^1) = 2$	$\#significant(n_0^2) = 0$	$\#significant(n_0^3) = 3$
$GrowingFactor(n_0^1, 1) = 1.1$,		$GrowingFactor(n_0^3, 1) = 0.08$,
$GrowingFactor(n_0^1, 2) = 1.5$,		$GrowingFactor(n_0^3, 2) = 0.11$,
$GrowingFactor(n_0^1, 3) = 0.8$,		$GrowingFactor(n_0^3, 3) = 0.26$,
$GrowingFactor(n_0^1, 4) = 0.7$		$GrowingFactor(n_0^3, 4) = 0.33$

Note that data sets are automatically classified by means of an algorithm that implements the categorisation presented in Section Hierarchical Data Categorisation.

The first condition to distinguish both tendencies checks the number of direct children of the root node, $width(n_i^k)$. This value fixes the tendency of the tree, which is elongated whenever this value is below the threshold value, N, and compact otherwise.

Moreover, the second condition checks the significance of the node n_i^k, that is if it has significant children, and its growing factor. It should be noted that when a node that has no direct significant children, its Growing Factor makes no sense, that is $\#significant(n_i^k) = 0$, Table 1(b) is an example of a news, node n_0^k in blue colour, that has no significant children. Thus, in that cases, the tendency is only based on the width of the node because $\#significant(n_i^k) = 0$ is true (see the first part of the second condition in Equations (4) and (5)).

The value L in the second condition models the proportion of the tree needed to determine its shape's tendency. For example, by setting L to 1, we are only considering the trend of the shape at the first level but not in the rest of levels, and by setting L at the maximum depth of the tree, we are demanding all its levels to strictly follow that tendency. Note that this constraint is theoretically possible but it is not easy to happen in real data sets. Actually, an intermediate value of L fixes the trend in the first L levels of the tree, without considering how the rest of the levels behave. Table 1 shows the $GrowingFactor(n_0^k, s)$, with s between 1 and 4 ($L = 4$) in columns a and c.

In addition, the value of the growing factor, GF, tell us about tree's growth, that is it determines how the L sublevels of a node maintain in relation to how it does. For example, if the same tendency is maintained in the successive L sublevels of the tree, then the *growingFactor* is approximately 1, which strictly guarantee the tendency of the whole tree. In the case of elongated tendency, to control how the tree expands we define a maximum value, $GF_{Elongated}$ below which we consider that the tendency is maintained (see Figure 2(a)). Similarly, we define a minimum value, $GF_{Compact}$ up to which we consider that the tendency is maintained in the case of compact tendency (see Figure 2(b)).

For instance, Table 1(a) shows a root node with 9 direct children (less than $N = 15$) and the following s levels up to $L = 4$ with a $(GrowingFactor(n_i^k, s) \leqslant GF_{elongated}(= 1.9))$. Thus, according Equation (4), we can affirm that the root node, n_0^k follows an Elongated Tendency (ET).

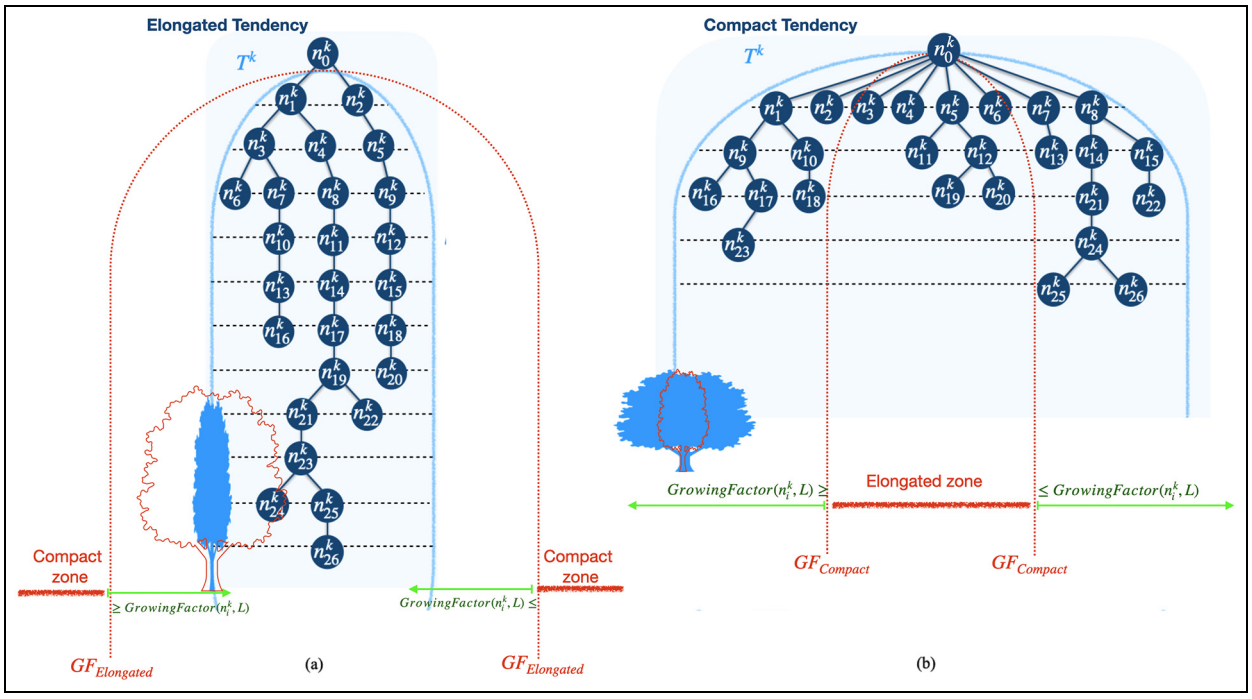


Figure 2. Overview of: (a) Elongated and (b) Compact tendencies. In red, zones where tendency changes from elongated to compact, and from compact to elongated.

Additionally, we can detect some parts of the hierarchies that become as linear sequences, that is, such spines. Then we can easily identify that a spine starts at node n_i^k when the relation between the number of nodes of the subtree of n_i^k, and the number of sublevels of that subtree is close to 1. Let denote $spine(n_i^k)$ if $\frac{\mathcal{D}(Subtree(T^k, n_i^k))}{\#Subtree(T^k, n_i^k)} = 1$.

Considering elongated and compact tendencies, as depicted in Figure 2, we propose the following categorisation of hierarchies:

1. T^k is Elongated when all the significant nodes through L levels starting at the root node of T^k fulfil the elongated tendency property, $ET(n_0^k, L)$ (see Figure 3). Notice also that spines are a particular case of elongated structures.
2. T^k is Compact when the main trend of the hierarchy fulfils compact properties at the root of T^k or at any node/s at some distance to it, D, and also it has not narrow (elongated) subtrees (see Figure 4).

$$
\begin{aligned}
&(CT(n_0^k, L, GF_{compact}) && \vee \\
&\exists n_i^k : CT(n_i^k, L, 1.0), distance(n_i^k, n_0^k) < D) && \wedge \\
&\nexists n_i^k : ET(n_i^k, L)\ \forall 0 \leqslant i \leqslant \#Nodes^k &&
\end{aligned}
$$

It should be noted that not all hierarchies fall into these two categories and thus may remain as Unspecified. Those Unspecified hierarchies may contain both elongated and compact subtrees, or several compact clustered regions. In this research, we focus on the study of the layouts that fit in well with Elongated and Compact ones.

Algorithm

In this section, we depict the main strategy followed to categorise the inner shapes of hierarchical data (see

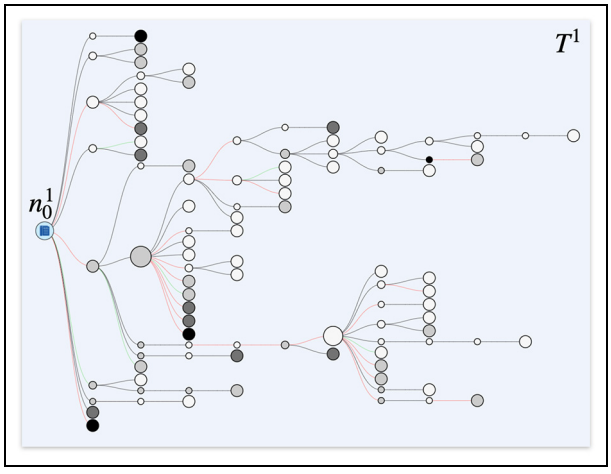


Figure 3. Example of Elongated hierarchy: $width(n_0^1) = 9$, $\#significant(n_0^1) = 1$, $GrowingFactor(n_0^1, 1) = 0.55$, $GrowingFactor(n_0^1, 2) = 1.33$, $GrowingFactor(n_0^1, 3) = 0.55$, $GrowingFactor(n_0^1, 4) = 0.66$, Threshold values: $N = 25$, $L = 4$, $GF_{Elongated} = 2$, $tolerance = 0.15$.

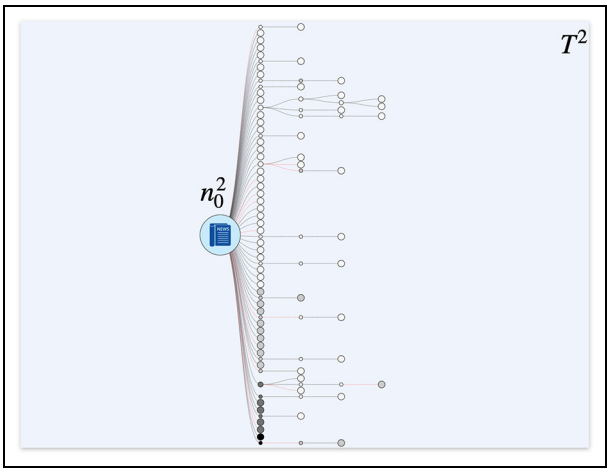


Figure 4. Example of Compact hierarchy: $width(n_0^2) = 60$, $\#significant(n_0^2) = 0$, Threshold values: $N = 25$, $L = 4$, $GF_{Compact} = 0.25$, $tolerance = 0.15$.

Algorithm 1). As inputs, it takes the entire hierarchy (T^k), the root node (n_0^k), and threshold values. It is noteworthy that n_i^k can be any node of the hierarchy. As we stated above, the algorithm uses the threshold values to check some properties of the hierarchy (N, L, D, $GF_{Elongated}$, $GF_{Compact}$, $tolerance$).

First, in lines 1–12, we check if the root node has Elongated Tendency (see $ET(n_0^k, L, GF_{elongated})$), verifying the conditions stated in Equation (4)). If so, we go through all the significant children (*getAllSignificant Children*()) to test their compactness (using the method $CT(n_i^k, L, GF_{compact})$). Only if no compact subtree exists, the hierarchy will be categorised as Elongated. Otherwise, the hierarchy is elongated in the first level but contain some compact structure and thus, it is categorised as Unspecified.

Secondly, in lines 13–29, the algorithm finds a node with Compact Tendency, n_c^k, (i.e. a node that fulfils the condition of Equation (5), $CT(n_0^k, L, GF_{compact})$). Note that this node can be directly the root node of the hierarchy, n_0^k, or any node close enough to the root node, $dist(n_i^k, n_0^k) < D$ (see lines 13–16). An then, similarly to the Elongated case, we traverse all the significant children (*getAllSignificantChildren*()) to test their elongation (using the method $ET(n_i^k, L, GF_{elongated})$). Again, only if no elongated subtree exists, the hierarchy will be categorised as Compact. Otherwise, the hierarchy is compact in the first level but contain some elongated structure and thus, it is categorised as Unspecified.

Finally, in case that the algorithm does not find elongated neither compact tendencies in the root node, we categorise the hierarchy as Unspecified.

Algorithm 1 Hierarchical data categorisation algorithm

Require: T^k, n_0^k, $GF_{elongated}$, $GF_{compact}$, $L \geqslant 0$, $D \geqslant L$, $0 \leqslant tolerance \leqslant 1$, N

```
 1: if ET(n_0^k, L, GF_elongated) then
 2:     sign = getAllSignificantChildren(T^k, n_0^k, L, tolerance)
 3:     isCompact ← False
 4:     for all n_i^k in sign do
 5:         isCompact ← CT(n_i^k, L, GF_compact)
 6:     end for
 7:     if isCompact then
 8:         Return Unspecified
 9:     else
10:         Return Elongated
11:     end if
12: end if
13: if CT(n_0^k, L, GF_compact) then
14:     n_c^k ← n_0^k
15: else if ∃n_i^k : CT(n_i^k, L, GF_compact), dist(n_i^k, n_0^k) < D
16:     n_c^k ← n_i^k
17: end if
18: if ∃n_c^k then
19:     sign = getAllSignificantChildren(T^k, n_c^k, L, tolerance)
20:     isElongated ← False
21:     for all n_j^k in sign do
22:         isElongated ← ET(n_j^k, L, GF_elongated)
23:     end for
24:     if isElongated then
25:         Return Unspecified
26:     else
27:         Return Compact
28:     end if
29: end if
30: Return Unspecified
```

Multivariate data categorisation

Next, we formalise the types of features contained in multivariate data. This formalisation will help us later in Section Visualising Multivariate Data to analyse the design elements (e.g. glyphs, icons) that best symbolise them. It should be noted that each node of the hierarchy contains data from which a set of *numF* predefined features, $\mathcal{F} = \{f_1, \ldots, f_{numF}\}$, will be extracted. We denote $data_{n_i^k}$ as the data relative to the node n_i^k. Analogously, we define $data_{e_{i,j}^k}$, and $data_{n_0^k}$ as the data contained in each edge $e_{i,j}^k$ of the tree, and in the root node n_0^k respectively. Thus, the total information stored in a tree, T^k, is

$$\mathcal{D}ata_{T^k} = \mathcal{D}ata_{Nodes^k} \cup \mathcal{D}ata_{Edges^k} \cup \{data_{n_0^k}\} \quad (6)$$

being

$$\mathcal{D}ata_{Nodes^k} = \{data_{n_i^k}\},\ \forall n_i^k \in Nodes^k,$$
$$\mathcal{D}ata_{Edges^k} = \{data_{e_{i,j}^k}\},\ \forall e_{i,j}^k \in Edges^k$$

Above we introduced our multivariate data categorisation neutrally without concretely basing it on any type of data. It should be noted that this categorisation can be applied to any multivariate data. Moreover, to explain our idea further we use our case study as an example below. As an example of a hierarchy T^k, the root node contains as $data_{n_0^k}$ the text of the news: *'A young North African is beaten after a violent robbery of an old woman'*. Then, a direct child n_j^k of the root node n_0^k will contain as data $data_{n_j^k}$ the message that replies to the news: *'A fucking piece of shit, he and those who lynch him, let's see if we understand that we live in a civilisation and not in the jungle. The thief is detained and the police are called'*. In this example, the edge between both nodes n_0^k and n_j^k does not contain any related data ($data_{e_{0,j}^k} = \emptyset$). However, when a direct child n_i^k of a node n_j^k, contains data, $data_{n_i^k}$, that relates to the data in $data_{n_j^k}$, the edge between them contains the data of both nodes: $data_{e_{i,i}^k} = < data_{n_j^k}, data_{n_i^k} >$.

Based on previous definitions, we state the labelling function, $\mathcal{L}$, as a function that associates a list of features to each element (either node or edge) of T^k according to its information:

$$\mathcal{L} : \mathcal{D}ata_{T^k} \longrightarrow f_1 \times f_2 \times \ldots \times f_{numF},\ f_i \in \mathcal{F} \qquad (7)$$

Moreover, we can define $\mathcal{L}$ separately for each type of tree element. Thus, we define $\mathcal{L}_{Nodes^k}$ and $\mathcal{L}_{Edges^k}$ with their related information, $\mathcal{D}ata_{Nodes^k}$ and $\mathcal{D}ata_{Edges^k}$ respectively.

$$\mathcal{L}_{Nodes^k} : \mathcal{D}ata_{Nodes^k} \longrightarrow f_1 \times \ldots \times f_{numF_{Nodes}},\ f_i \in \mathcal{F}_{Nodes}$$

$$\mathcal{L}_{Edges^k} : \mathcal{D}ata_{Edges^k} \longrightarrow f_1 \times \ldots \times f_{numF_{Edges}},\ f_i \in \mathcal{F}_{Edges}$$

It is worth noting that each dimension, or feature, of $\mathcal{F}$, f_i, defines a variable in the domain that can be numerical (discrete or continuous) or categorical (nominal or ordinal). Some of them are independent variables, but others are dependent allowing to model cause-and-effect relationships.

For instance, in our case study the set of features associated to the tree nodes, $\mathcal{F}_{Nodes}$, is a set of categorical features that define the spectrum of the speech related to the comments, such as constructiveness, argumentation, sarcasm, mockery, insult, improper language, intolerance, aggressiveness, target person, target group, stereotype, toxicity and the level of toxicity, that correspond to $f_1, f_2, \ldots, f_{13}$, respectively. Some of these features are nominal features representing two values, such as f_1 (a message is constructive or non-constructive), and some others include ordinal features, such as f_{13} that represents the four levels of toxicity – not toxic, mildly toxic, toxic and very toxic –.

Additionally, the feature relative to the edges of the tree is related to the stance of a comment in relation to the previous one, and thus, $\mathcal{F}_{Edges} = \{f_{14}\}$, that is f_{14} represents the stance, that is also a nominal variable representing three values – the stance of a message can be positive if it reinforces the opinion of the previous message, negative, if it is against, and neutral, otherwise –.

Following the example introduced above about a news and a possible reply, $data_{n_j^k}$, the labelling function of the reply produces the following result: $\mathcal{L}_{nodes}(data_{n_j^k}) = <$ `not constructive, argumentative, not sarcastic, not mockery, not intolerant, improper language, insult, not aggressive, target person, no target group, no stereotype, toxic, mildly toxic` $>$, and $\mathcal{L}_{edges}(data_{e_{0,j}^k}) = \emptyset$.

And, for example, when a node n_i^k supports the opinion of its father n_j^k, then the label of the edge between them is defined by: $\mathcal{L}_{edges}(data_{e_{j,i}^k}) =$ `positive stance`.

Additionally, we group related features in *NumC* clusters depending on their semantics, $\mathcal{C} = \{c_1, \ldots, c_{NumC}\}$, where $c_i = \{f_1^i, \ldots, f_{numF^i}^i\}$, being $\forall f_s^i \in \mathcal{F}$, $1 \leqslant s \leqslant numF^i$.

For example, in our case study, the cluster c_1, includes three features that refer to the targets the comment focuses on, $c_1 =$ {`target person, target group, stereotype`}.

Actually, our hypothesis H3 is based on these clusters to find out the best pictorial representation for features and where to locate it on the chosen layout, according to the nature of the features themselves.

Thus, in the next section, we analyse the expressiveness and comprehensibility of different layouts that display overview visualisations of hierarchical structures, that is the whole tree structure. Then, we detail the graphical elements chosen to represent the different types of features (multivariate data), and their relationship with the hierarchical layouts.

Hierarchical visualisation layouts

The literature offers a huge visual bibliography of hierarchical visualisations,[9] that includes 333 techniques, which makes it challenging to find the best-fitted visualisation for different hierarchical structures. It should be pointed out that most of these techniques are derived from each other by adding additional features or advancing the existed techniques. For example, Multivariate Bubble Treemap by Zheng and Sadlo[35] has the same visual as Bubble Treemap by

Gortler et al.[36] but it includes additional glyphs. Moreover, some of these techniques (e.g. radial, tree, etc.) are the most used.[37] In the following, we analyse the most common implicit and explicit hierarchical layouts to set the scope of our analysis.

The use of implicit layouts,[13] such as treemaps, Circle packing, Icicle or Sunburst diagrams, where the parent-child relationships are coded using relative locations between parents and children, are space efficient due to their high compactness. However, it is harder to read huge-sized, broad and deep hierarchies with these layouts. In addition, as these layouts place nodes in a nested way without leaving empty spaces, the inclusion of icons and glyphs representing Multivariate features becomes more difficult.

We illustrate these ideas by displaying an online news article and its comments using the Circle packing layout in Table 1. This layout is the circular version of a treemap where nodes are packed in circles. The root node (e.g. news article) is represented as the biggest circle that contains all the nodes (see the big blue circle in Table 1(a)). Direct children of a node are placed inside the circle relative to their parent node. The more children a node has, the larger is the circle is.

While its compactness is an advantage with small-sized data that has few levels, it can be a disadvantage, especially with broad data that has most of its nodes on the same level as it becomes overcrowded easily (Table 1(b)). Moreover, where siblings with a different number of children, the circle packing uses different sizes in the same levels, losing the perception of the relationships between them. On the other hand, in narrow and deeper hierarchies, the nested circles make it difficult to understand the hierarchy structure (Table 1(a)). Moreover, long spines (i.e. large narrow branches) are visualised as many concentric circles, because of the consecutive placement of children onto their parents' circles, thus, making it hard to appreciate the different levels of the hierarchy and the parent-child and siblings relationships (see Table 1(c)). In addition, if we also add pictorial representations of the features, the visualisation ends up being even more crowded. In this example, there are nodes in which it is hard to discern their level of toxicity even taking into account that it only shows 1 to 4-valued feature (the level of toxicity), using just a colour range (white for non-toxic comments to black for high toxic comments).

Unlikely to treemaps, Sunburst diagram[38] and Icicle plot[39] implicit layouts show the parent-child relationships by placing the child nodes next to their parents nodes, circularly in Sunburst and linearly in Icicle plot. While these two layouts could better show the hierarchy than treemaps, and use the space more efficiently, they will have similar problems displaying large-sized data that has long spines. Especially, when the data is big the very outer leaves on hierarchies are displayed as very thin rectangles on both layouts thus, this will make the graphs harder to be analysed in a complete view. Additionally, in Icicle plot the same level nodes are placed next to each other in a horizontal line and when we consider broad data that has a compact tendency, the visualisation has to be either horizontally extended beyond the screen, or shrunk with zoom-out, in both ways it loses the global view. While due to its circular shape Sunburst will not have this problem, when the data is too dense it can become crowded easily, and an overcrowded visualisation hinders the effective exploration of the data. Moreover, if multiple attributes are integrated with these visualisations, they will be harder to read and analyse on overview and it would be impossible to see multivariate attributes on the slimmer nodes.

On the other hand, explicit layouts,[9] that is node-link layouts (tree, radial, force-directed), have better readability over viewing hierarchies as each node is shown individually.[40] However, they are also known for using space not very efficiently due to the lines that connect parent and children nodes occupying space and generating empty backgrounds. Indeed, this can be an advantage while visualising multivariate attributes as there is plenty of space to map additional elements such as glyphs. Also, they can visualise data both on the nodes and edges. In the following, we show three data sets (elongated, compact and unspecified) using review explicit layouts in Figures 5–7.

The Tree layout in Figure 5 is laid out horizontally. Thus, the depths of the nodes are shown horizontally and, all nodes in the same depth are placed in the same vertical line. Particularly, the tree layout is well organised to visualise narrow structures, it clearly presents the relationships between siblings in different levels (see Figure 5(a)). However, if a hierarchy has nodes with a lot of direct children on the same level, they will be placed on the same vertical line forming a very long straight column with very small nodes (see Figure 5(b)), losing details and wasting canvas space. Especially, when the wide data is also crowded in all levels tree layout loses its comprehension and becomes difficult to visualise the entirety of the structure at once (see Figure 5(c)).

Radial layout (see Figure 6) arranges nodes on concentric circles. It is better than the tree layout when the data is broad since it uses space more efficiently by arranging hierarchies circularly. Thus, the Radial layout fits larger amounts of nodes into the canvas (see Figure 6(b)). For whenever its circles are partially filled, human perception through the Gestalt's principle of closure,[41] could reconstruct them as long as they are sufficiently populated with nodes that are evenly distributed in each level such as in compact

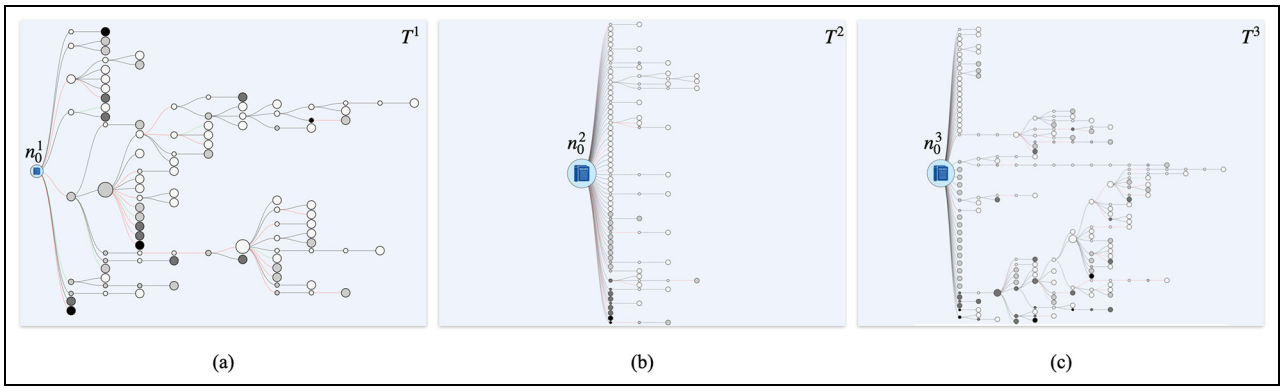


Figure 5. Tree layout with the same data sets shown using Circle Packing, properties (significant, growing factor, etc.), and threshold values shown in Table 1: (a) data set categorised as Elongated, (b) data set categorised as Compact and (c) unspecified data set.

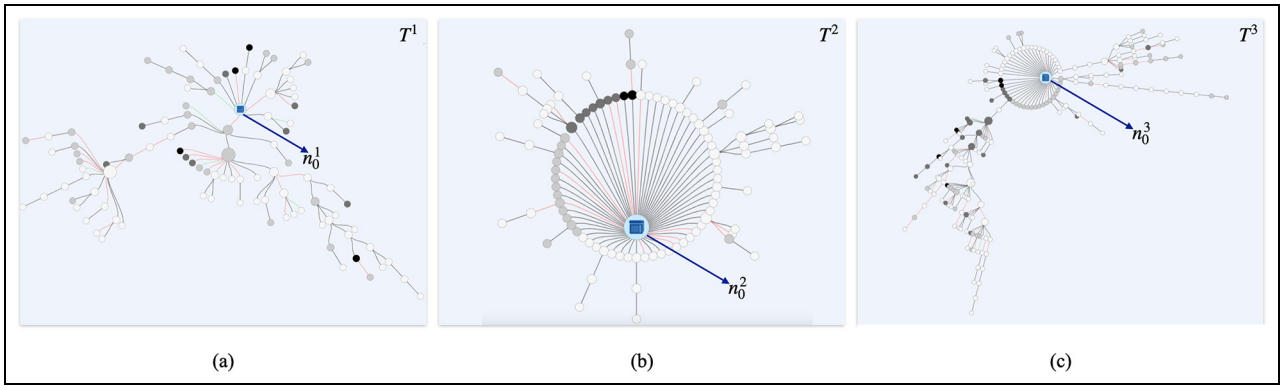


Figure 6. Radial layout with the same data sets shown using Circle Packing, properties (significant, growing factor, etc.), and threshold values shown in Table 1: (a) data set categorised as Elongated, (b) data set categorised as Compact and (c) unspecified data set.

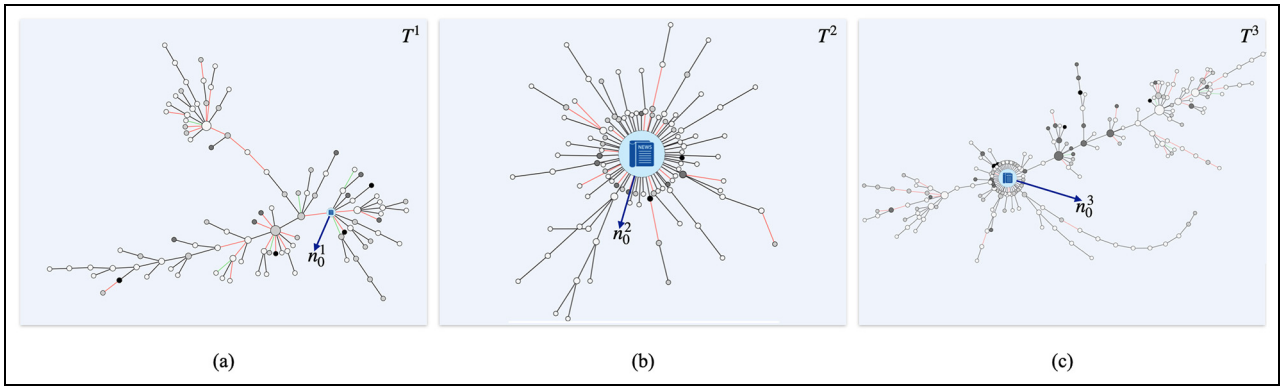


Figure 7. Force layout with the same data sets shown using Circle Packing, properties (significant, growing factor, etc.), and threshold values shown in Table 1: (a) data set categorised as Elongated, (b) data set categorised as Compact and (c) unspecified data set.

data. Thus, the radial layout offers a comprehensible visualisation of hierarchies with compact data as the total number of nodes in each level is in proportion to the number of nodes in other levels (see Figure 6(b)). Otherwise, when data has elongated tendency characteristics, the perception of closure is lost (see Figure 6(a)). Moreover, a hierarchy can initially have a perception of closure in the first few levels however, this perception can be lost if the rest of the hierarchy does not follow the initial tendency, such as by having long threads away after compact first few levels (see Figure 6(c)).

Force layout (see Figure 7) also displays the hierarchy somehow in a circular way. However, unlike Tree and Radial layouts, it does not place the same depth nodes in an ordered alignment, thus it is not as effective as showing the relationship between sibling nodes but it gives nodes more freedom on the canvas. Also, due to its force-based strategy that uses energy functions to place nodes in the canvas,[42] force layout visualises groups (threads) of data in clusters and places them away from each other. Thus, the Force layout could efficiently display broad (see Figure 7(b)) and large-sized (see Figure 7(c)) hierarchies in a global view. However, in Figure 7(a) there is a Force layout with narrow data we can observe that it is difficult to appreciate the relationships between siblings due to the uneven distribution of nodes in each level.

In this research, in order to compare different hierarchical layouts, we selected the Circle packing as the representative of the implicit layouts and the Tree, Radial and Force layouts as the representative of the explicit ones. Circle packing stood out as it was the most different among others. Due to their shapes, Sunburst and Icicle plot looked somehow similar to our explicit layout selections, Radial and Tree respectively. Regarding explicit layouts, Tree and Radial layouts are the baseline displays for narrow and broad data, respectively. We also selected Force layout as it may arrange nodes dynamically taking into account available space in the canvas.

Visualising multivariate data

Multivariate data implies visualising a high amount of features without overwhelming users' perceptions. There are wide range of approaches in the literature proposing different visual elements and techniques to communicate a high number of features at once, such as colours, shapes, icons and glyphs. Although *one-by-one* based approaches depict features side by side,[11] *all-in-one* approaches group together interrelated features.[12]

Multivariate data visualisations are challenging in themselves but even more when they should be integrated into hierarchical visualisations. On the one hand, implicit hierarchical layouts, due to their high compactness use space efficiently but are left with little empty space to integrate visual elements representing multivariate data. On the other hand, the low compactness of explicit layouts can be turned into an advantage, as the empty space, and also edges could be used for visualising features. Concretely, in this research, we visualise multivariate data using explicit layouts (Tree, Radial, Force).

Firstly, let's consider ordinal data with a list of values, for instance, a three-valued size feature (small, medium, large), and each node has one of those different values. As these kinds of data should be represented in every node, the best way to visualise them is directly on the visualisations. For example, they could be directly shown on nodes or edges as hue colours. As usually ordinal features have more abstract meanings, using icons is not ideal. Moreover, using shapes fixed next to each other on the visualisations can complicate them very easily. If the node contains more than one ordinal feature or one ordinal feature combined with other types of features then glyphs can be also an option.[29]

Secondly, regarding nominal features (e.g. hair colour blond – yes or no –), to not overwhelm the user perception, only those nodes or edges that fulfil the property will display its visual element. For example, only nodes that represent blond people will show the visual representation of blond hair. Moreover, features with concrete meanings would be well represented with icons. For example, if recycling is the tagged feature it could be easily visualised with a recycle icon. On the other hand, for more abstract features such as being sarcastic and intolerant, glyphs could be a good option and each feature can be mapped onto this glyph, with unique hue colours, without any additional symbol or icon.

In the following, we use the cluster-based multivariate data categorisation presented in Section Multivariate data categorisation, to analyse those design elements that could be combined to visualise multivariate data. One option to visualise clusters of features is to use *one-by-one* glyphs, that is, visualising them next to nodes either linearly, ordered one next to each other, or circularly around the node, which could be very compatible with tree and radial layouts as these layouts are more structured (see Figure 8(a) and (b)). However, as Force layout distributes its nodes more freely, the linear placement of *one-by-one* glyphs could look confusing and hard to detect which glyph belongs to which node and, then the option to place

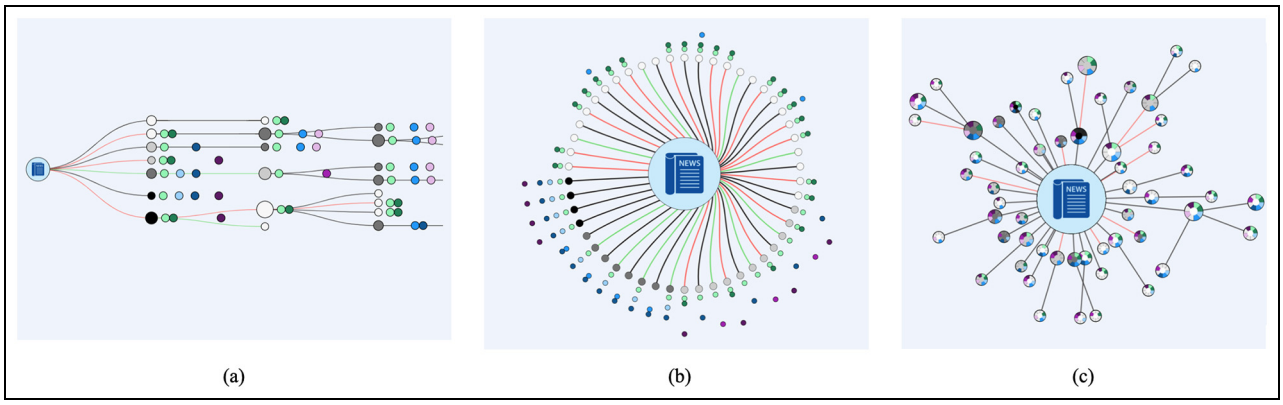


Figure 8. *One-by-one* glyph shown on: (a) tree layout, (b) radial layout and *all-in-one* glyph shown on (c) force layout.

circularly. Another option for force layout could be visualising glyphs as *all-in-one* on the nodes as shown in Figure 8(c). Note that *all-in-one* glyphs should accommodate all the features in the cluster in less space than in the *one-by-one* case. Additionally, *all-in-one* glyphs could be either placed outside or inside the nodes. In the latter, the space devoted to each feature will be even smaller in size as it depends on the size of the nodes. Thus, *all-in-one* glyph could be more useful on a detailed view while *one-by-one* could be a more helpful method while visualising the global view of the hierarchies.

Case study

Our research uses a specific data model that has been developed by our linguistics team[43] which aims to study the hate speech occurrence in the news comment sections. The data was collected from different online news articles' comment sections (e.g. El Mundo). Then, the corpus was developed by three annotators and a senior professor who was an advisor in the annotation process, this process is thoroughly explained in Kavaz et al.[44] The corpus consists of 4359 comments posted in response to news articles extracted from online newspapers from August 2017 to July 2020 annotated with toxicity. The articles were selected to cover three different topics that are immigration, society and crime. Table 2 shows the distribution of comments per topic and the number of news articles in each topic. News articles had minimum of 60 comments and maximum of 360 comments.

Detection of hate speech is a difficult task because it has a highly and inevitably subjectivity. Thus, linguists established a specific tag set features with subproperties to help them to determine the degree of hate-speech in the comments by measuring toxicity in comments. The data is annotated with 14 features in the annotation process (e.g. constructiveness, sarcasm, etc.), we introduced all the features in the example of the multivariate data formalisation in Section multivariate data categorisation Thus, our data is hierarchical and multivariate.

Table 2. Distribution of comments per topic.

Area	Comments	No of news articles (T^k)
Immigration (IM)	1651	8
Society (SO)	866	5
Crime (CR)	1842	8
Total	4359	21

To visualise each multivariate feature in the best way we used our formalisation to cluster these features according to their characteristics. Thus, we now present the clusters and the visualisation methods we used for each one.

Cluster 1, c_1 includes an ordinal feature, level of toxicity, c_1 = {`0 (non-toxic), 1 (mildly-toxic), 2(toxic) and 3 (very toxic).`}. As this is an ordinal feature and each node is tagged with one of the values the best way to visualise this feature is as hue colours on the nodes. We used a colour range from white to black to represent non-toxic to very toxic. Moreover, the level of toxicity is the most important feature in our case study thus, that would match our goal to visualise it in the global view of hierarchies as this will give information at the first glance.
Cluster 2, c_2, includes nominal features that can be represented with edges, c_2 = {`neutral stance, positive stance, negative stance`}. As these features are related to the edges the best option is to directly show them on the layouts as hue colours. Neutral, positive, and negative stances are represented with black, green and red respectively. Moreover, if a node has both positive and negative stances it is shown as orange.

Cluster 3, c_3, represents nominal features that have concrete meanings, c_3 = {`target person`, `target group`, `stereotype`}. Thus, we created an icon for each feature in this cluster and showed them on-demand next to the nodes that have these features (see Figure 9(c) and (d)).

Finally, *Cluster 4*, c_4, represents eight nominal features with abstract meanings, c_4 = {`constructiveness`, `argumentation`, `sarcasm`, `mockery`, `intolerance`, `improper language`, `insult`, `aggressiveness`}. Therefore, we created two glyphs to visualise these features: (i) an *one-by-one* glyph (see Figure 9(a)), where features are represented by coloured dots that are placed next to the nodes in an ordered row (Figure 8(a) shows an example) and (ii) an *all-in-one* glyph (see Figure 9(b)), placed on the node, depicted as a pie chart including eight equal pieces with unique hue colours for each feature, displaying the level of toxicity of the nodes on its centre. Both glyphs used green shades for more positive features (i.e. constructiveness), blue for more neutral features (i.e. sarcasm), and magenta for more negative features (i.e. insult). Similarly to *Cluster 3*, these glyphs will be shown on-demand.

Evaluation

We conducted a user evaluation to study our hypotheses. Thus, the purpose of the evaluation was twofold: (1) to validate the best fitted visualisation layouts according to the proposed categories of hierarchies, Elongated and Compact and (2) to determine which glyph (*one-by-one*, *all-in-one*) is more useful with layouts while visualising data globally.

Methodology

We recruited 35 participants of which two of them were lecturers while the rest were students from the Faculty of Philology and Communication at the University of Barcelona; 60% of the participants were female, 80% were aged between 20 and 30 and 17% of the participants had experiences in message annotation and data visualisation but in general, participants had no experience in these fields. The study was a within-subject and a moderated test, conducted in a classroom.

At the beginning of the evaluation session, we explained to participants our study, the structure of conversation threads, annotations (tagged features), and then we briefly explained how they were mapped onto our visualisation layouts. We aimed to facilitate the understanding of the tagged attributes by explaining this in more depth and to make sure that users focus on the visualisations rather than trying to understand how annotations worked. Afterwards, we gave them a couple of minutes to engage with the visualisations before we started the test to ensure that our users understood the tool. Also, before each task we gave users tips regarding how to use our visualisation tool (e.g. showing glyphs) that could help them solve the tasks, again, to make sure that users focus on the visualisations rather than trying to understand tool's functioning. There were a total of five tasks (see Table 3, where EC and CC refer to Elongated Categories and Compact Categories, respectively), to make sure that there was no learning effect in performing the first 4 tasks we prepared two versions, half of the users did Test A with the task order T1-EC, T2-EC, T3-CC, T4-CC, T5, and the other half did Test B with the task order T3-CC, T4-CC, T1-EC, T2-EC, T5.

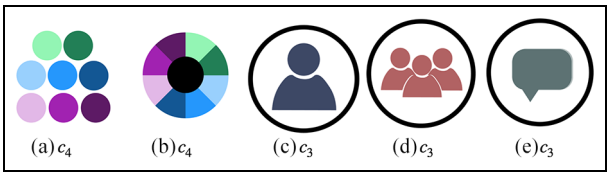


Figure 9. Glyphs: (a) one-by-one and (b) all-in-one and target icons: (c) target person, (d) target group and (e) stereotype.

Research hypotheses and associated tasks

We aimed to analyse our hypotheses with tasks that are defined from linguists' research questions. Each task is assigned with a chosen data set according to the Elongated and Compact categories.

Tasks 1 and 2 were designed to explore ***Hypothesis 1***: *'When the hierarchy has Elongated Category the most informative layouts to visualise it are Tree layout and Circle Packing'*. To test our hypothesis, in Tasks 1 and 2 we selected data sets that are Elongated thus we will refer to these tasks as T1-EC and T2-EC. In T1-EC we asked users to only engage with Circle Packing and Tree layout, and in T2-EC we asked users to play with all layouts. In this way, in T1-EC we could compare the two layouts we selected for the Elongated category to see if they are equally useful for the users to perform the visualisation tasks. Moreover, with T2-EC we could compare our selections (Tree and Circle) for the Elongated category with users' selections as we expect them to have higher scores than the other layouts (Force and Radial) in this category.

Task 3 and Task 4 were designed to explore ***Hypothesis 2***: *'When the hierarchy has Compact Category the most informative layouts to visualise it are Force layout and Radial layout'*. Thus, we assigned data sets with the Compact category to these tasks and we

Table 3. Evaluation: research hypotheses and their associated tasks, layouts, data sets and data gathered on each task.

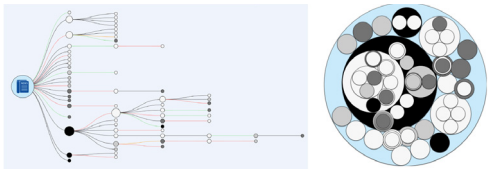
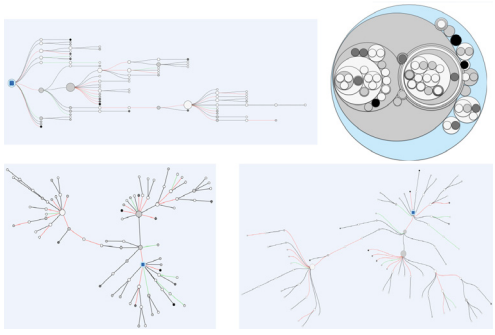
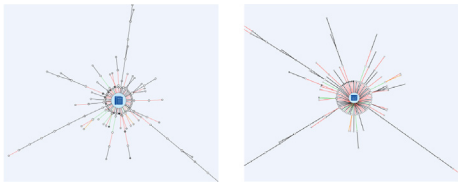
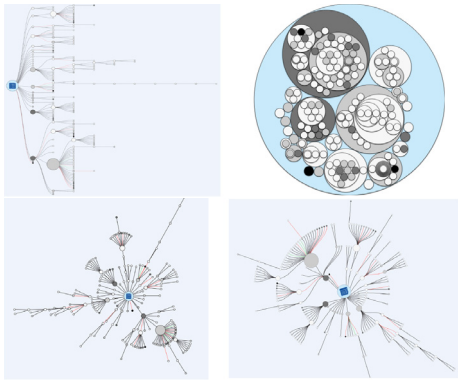
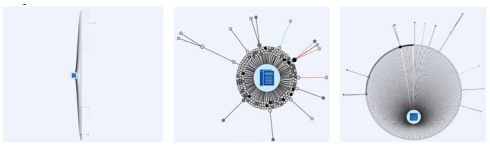
Research hypotheses	Task	Visualisation layout	Data set	Data gathered
H1 Elongated Category	**T1-EC:** In Comment level 2, in which Level of Toxicity does the combination of Intolerance and Stereotype comments appear more?	Tree layout and circle packing 	Elong. News Article IM_1	Ratings of two layouts
	T2-EC: In Comment level 2, in which Level of Toxicity does the combination of Improper Language and Insult comments appear more?	Play with all four layouts 	Elong. News Article: SO_1	Ratings of four layouts
H2 Compact Category	**T3-CC:** In Comment level 1, in which Level of Toxicity does the combination of Target Group and Mockery comments appear more?	Force layout and Radial Layout 	Comp. News Article: SO_2	Ratings of two layouts
	T4-CC: In Comment level 2, in which Level of Toxicity there are more Aggressive comments?	Play with all four layouts 	Comp. News Article: IM_2	Ratings of four layouts
H3 Glyphs	**T5:** Which Features appear more with Target Group in Level of Toxicity 3 (very toxic)? In this task please use *one-by-one* glyph, *all-in-one* glyph and Target Icons	Layouts - Tree, Force and Radial 	Comp. News Article: CR_1	Ratings of two glyphs

Table 4. Hypothesis 1, Elongated categories (paired t-test).

Layouts (Mean, SEM)			p-Value	ES
T1-EC	Tree (4.17, 0.14)	Circle (2.77, 0.23)	< .001*	0.85
T2-EC	Tree (4.17, 0.18)	Force (3.09, 0.20)	< .001*	0.74
	Tree (4.17, 0.18)	Radial (2.74, 0.20)	< .001*	1.00
	Circle (2.74, 0.22)	Force (3.09, 0.20)	0.195	—
	Circle (2.74, 0.22)	Radial (2.74, 0.20)	1.000	—

*significant at the $p < 0.05$ level.

Table 5. Hypothesis 2, Compact Structures (paired t-test).

Layouts (Mean, SEM)			*p*-Value	ES
T3-CC	Force (3.77, 0.18)	Radial (3.65, 0.17)	0.607	—
T4-CC	Force (3.69, 0.21)	Tree (4.03, 0.19)	0.195	—
	Force (3.69, 0.21)	Circle (2.51, 0.23)	< .001*	0.83
	Radial (3.4, 0.21)	Tree (4.03, 0.19)	0.012*	0.45
	Radial (3.4, 0.21)	Circle (2.51, 0.23)	0.003*	0.54

*significant at the $p < 0.05$ level.

Table 6. Elongated versus Compact (paired t-test).

Layouts (Mean, SEM)		*p*-Value	ES
T2-EC	T4-CC		
Force (3.09, 0.20)	Force (3.69, 0.21)	0.016*	0.43
Radial (2.74, 0.20)	Radial (3.4, 0.21)	0.019*	0.42
Circle (2.74, 0.22)	Circle (2.51, 0.23)	0.530	—

*significant at the $p < 0.05$ level.

will refer to them as T3-CC and T4-CC. Similarly, in T3-CC we asked users to only engage with selected layouts, Force and Radial and in T4-CC we asked users to play with all layouts. Thus, we could compare if Force and Radial are equally sufficient in T3-CC. Furthermore, we wanted to compare if users agree with our selections as we expected that Force and Radial would score higher than Tree and Circle Packing in T4-CC.

Finally, Task 5 was designed to explore ***Hypothesis 3***: *'When a cluster contains a large number of features, the most informative representation to embed them in a hierarchical layout is an one-by-one instead of an all-in-one glyph'*. In this task, we asked participants to interact with both glyphs with the layout of their choice and rate considering the global view of the visualisation. We decided to use Compact data with Task 5 as we wanted to see if our glyphs are useful even with broad data sets.

In all tasks, we asked users *to rate the layouts they used to perform the tasks on a scale of very difficult (1) to very easy (5), N/A if applicable, and to write their comments, if they have any.* It should be reminded that in Task 5 we asked participants to play only with layouts Tree, Force and Radial as these are the visualisation layouts that support glyphs. Additionally, in Task 5, we asked users *to rate the glyphs they used to perform the task on a scale of not useful (1) to very useful (5) and N/A if applicable.* At the end of all tasks we asked about their overall comments to collect some qualitative data.

Results

We considered our sample size enough for computing statistical significances, as stated by.[45] Thus, we conducted paired t-tests and computed the Effect Size (ES) after rejecting the null hypothesis to measure the magnitude of mean differences (see Cohen's *d* ES in Tables 4–6). Notice that we obtained large effect size in most of the cases (Cohen's *d* ES > 0.8), meaning that means are likely very different. In the following, we present our results.

Hypothesis 1. Tasks T1-EC and T2-EC were designed to study Hypothesis 1 (elongated structure). Specifically, T1-EC was designed to compare Tree layout and Circle packing. As it can be observed in Figure 10 and Table 4, Tree layout (4.17 out of 5) received much higher scores than Circle packing

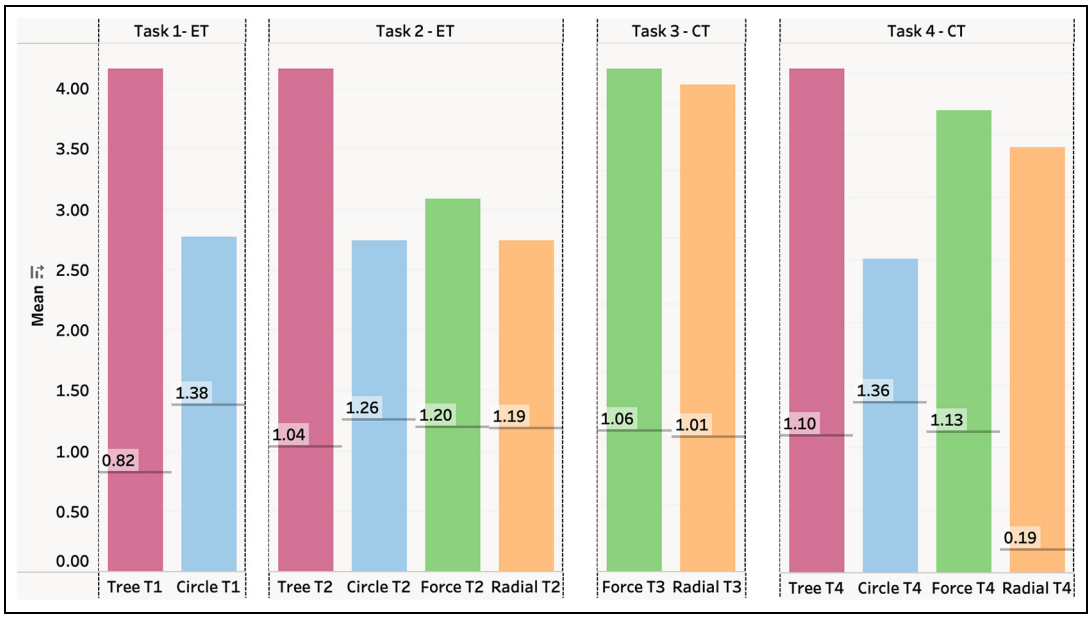


Figure 10. Mean values of layouts from Task 1 to 4 with standard deviation.

(2.77), the standard paired t-test shows a significant difference between their scores ($p < 0.001$). However, this may be as a result of participants' familiarity with Tree layout as hierarchical tree diagrams are being used in a variety of fields such as management planning[46] or diagramming sentences in linguistics.[47] Circle packing has a relatively different design that participants required more explanations about how conversation thread structures are mapped onto its circular design. Moreover, the results of T1-EC and T2-EC, show that Tree layout kept its popularity and has the same mean (4.17) in both tasks.

According to Hypothesis 1, we expected that Tree layout and Circle packing would score higher than Force and Radial layouts in T2-EC, recall that users played with all layouts in this task. When we compared the results of the Tree layout versus the Force and Radial layouts in T2-EC we found that the Tree layout (4.17) scored significantly higher than the Force (3.09) and Radial (2.74) layouts. The standard paired t-tests are proving the significant difference between Tree versus Force and Radial layouts as they are both $p < 0.001$. However, when we compared Circle packing (2.74) with the Force and Radial layouts in T2-EC with the standard paired t-test we couldn't find any significant differences. As we stated previously, this can be because of participants' unfamiliarity with a circular design of threads of comments.

Remark 1. *The results indicate that the most preferred layout with Elongated data is the Tree layout. Thus, H1 is partially supported by our results.*

Hypothesis 2. Tasks T3-CC and T4-CC were designed to study Hypothesis 2 (compact structure). As it can be seen in Figure 10 and Table 5, Force and Radial layouts in T3-CC scored relatively similar as expected, 3.77 and 3.65, respectively, and the t-test shows that their scores do not have a significant difference.

According to Hypothesis 2, we expected that the Force and Radial layouts would score higher than Tree and Circle in T4-CC. When we conducted the standard paired t-test between the Force (3.69) and Radial (3.4) layouts against Circle packing (2.51), both showed that they scored significantly higher than the Circle packing. The results of the t-tests of the Force layout vs Circle packing is ($p < 0.001$) and the Radial layout versus Circle packing is ($p = 0.003$). However, neither Force nor Radial layouts scored higher than the Tree layout. The reason, again, for this can be that the Tree layout is a more common visualisation layout when it is compared to others. Moreover, another reason for this may be because originally conversation threads are formed from top to bottom and in Tree layout, this is simply changed to the left to right. However, in other layouts the structure changed into circular designs, this caused participants to understand tree layouts more easily, therefore giving it higher scores.

Remark 2. *The results indicate that the most preferred graph with Compact data is Tree layout.*

Although the Tree layout was the best-scored layout in the previous analysis of Hypothesis 1 (elongated

Figure 11. Mean values of Glyphs from Task 5 with standard deviation.

data) and Hypothesis 2 (compact data), we believed that Force and Radial should score higher with Compact data and Circle packing should score higher with Elongated data. Then, we compared the scores of Force, Radial and Circle Packing layouts between T2-EC and T4-CC (see Table 6).

The results show that Force and Radial layouts scored significantly higher in T4-CC ($p = 0.016$) and ($p = 0.019$), respectively. Even though Force and Radial layouts scored lower than Tree layout in both tasks, we can see that with Compact data they scored significantly higher. Therefore, our hypothesis 2 can be partially supported. Furthermore, Circle packing received a higher score in T2-EC compared to T4-CC which is aligned with our Hypothesis 1. However, the difference is very little and we couldn't find any significant difference.

Remark 3. *Even though Tree layout was selected best with Compact data, Force and Radial layouts showed some evidence that they are more appropriate for Compact data rather than Elongated ones. Thus, this evidence confirm that our second hypothesis can be further explored. Even though it is not significant Circle packing received higher scores with Elongated data as expected.*

Hypothesis 3. Task 5 was designed to examine Hypothesis 3. When we compared the scores of *one-by-one* (3.91) and all-in-one glyph (3.48) we discovered that *one-by-one* scored higher as expected. (See figure 11) However, a paired t-test showed no statistical difference between them. Also, some users commented that the *one-by-one* option was more useful. This can be because the coloured dots used in *one-by-one* glyph can better scale with zoom-out when compared to the *all-in-one* glyph. The latter glyph can be better in a detailed view. As both received quite high scores we can conclude that our users find Glyphs useful.

Remark 4. *The results showed that participants preferred one-by-one over all-in-one glyph on the global view which aligns with H3.*

Qualitative results

We gathered some information from our open questions. Users' comments aligned with our Hypothesis 1. For example, user 20 stated that in T2-EC *'in tree layout it is much easier to detect the comment level of each comment'*, however, he also commented in T4-CC *'it is difficult to obtain a nice general perspective from all comments from tree layout'*. As compact data has a wider and more dense set of comments it can be harder to analyse them with tree layout.

While most users commented that Circle packing can be confusing and hard to understand, some users commented that it gets easier to understand as you get familiar with it. User 4 commented that *'Even though the tree, force and radial layouts seem easier to understand at first, I found that once you get the hang of it, the circle packing was the most useful layout to answer task 4'*, and user 10 stated that *'the circle packing layout seemed the most visually useful to me, but I needed a brief explanation to identify which circles were comment level 1 so as not to confuse them'*.

Also, we discovered that some participants needed more time to understand Force and Radial layouts. For example, when we follow the comments of user 11, in T1-EC and T2-EC the user commented that only the Tree layout was easy to understand and others, especially Circle packing, was very difficult. Then in T3-CC, the same user stated that *'in this task, I found it easy to use both display options (force and radial) in fact, I have a better idea of how they work'*. Finally, in T4-CC the user also found circle packing easier to use as the user got familiar with it.

Remark 5. *Although tree layout seemed the most intuitive for the users at the first glance, the rest of the layouts have a potential once the users know how to interpret them.*

Discussion

Previous studies,[13,14] gave some advice on the hierarchical visualisation methods suitable for some types of data and tasks. Nevertheless, they did not take into account the shape of the hierarchy dictated by the data, that is the inner structure of data. Other works in the literature visualised hierarchical data using a concrete layout. For example,[15] visualised their omics data using a radial layout,[48] used tree layout to visualise their ancestral data and,[17,19] which dealt with similar data to our research, used a radial layout too. Our findings, however, suggest that a categorisation of hierarchical data informs the visualisation method (layout) that best fits an overview of the data. Moreover, our

study opens the avenue of analysing whether it is adequate to change the layout depending on the tendency of the sub-trees resulting from some queries or selections. For instance, changing from radial layout to tree layout when the user navigates from a global view categorised as compact, to a local view of some part of the tree categorised as elongated. This same idea can be applied to create responsive hierarchical visualisations, which would change depending on the size of the device as[49] proposed to effectively present information based on the device context. Furthermore, this idea can be integrated when the size of the data is huge. For example, this huge data can be divided into regions and classified separately with our categorisation. Thus, we can visualise these regions using a combination of layouts.

Our categorisation is based on some features of the hierarchical data to detect Elongated and Compact tendencies starting from the node n_i^k, such as the significance of the nodes $(significant(n_i^k))$, how they grow ($width(n_i^k)$, and $GrowingFactor(n_i^k, s)$) and the scope of growth (controlled by the analysis of L levels). To do so, we needed some threshold values: for computing significant nodes, *tolerance*; for the number of direct children (width of n_i^k), N; for calculating the tendency along with a number of levels, L and for the maximum and minimum values of the growing factor of elongated and compact structures, $GF_{elongated}$ and $GF_{compact}$, respectively. It should be noted that these threshold values may depend on the data, especially determining the N value, which indeed helps to establish some kind of borderline between Elongated and Compact structures. We think that the values of these thresholds deserve further study, discarding then constant values such as L (in our study set to 4) and considering, for example, their computation based on some percentage of nodes of the tree. In this research we mainly focused on the categorisation of hierarchical data independently of the canvas dimensions. Nevertheless, the value of N (in our study set to 15) could be based on computing the canvas aspect-ratio what would categorise a same hierarchy differently depending on this aspect-ratio instead of the inner structure of the data.

Indeed, we proposed a formalisation to categorise hierarchical structures as Elongated or Compact, but there are some hierarchies that do not belong to neither, which we defined as Unspecified in our current categorisation. Thus, our classification can be further extended as we detected that some hierarchies include characteristics from both Elongated and Compact (Hybrid) and some of them may have more than one compact structure (N-compact). Indeed, we are investigating layouts that fit in well with these additional categories. For example, we think that Force layout could be a good option for visualising Hybrid and n-Compact hierarchies, as it gives nodes more freedom on canvas.

Although our results showed that Tree layout was selected by the majority of users as the most intuitive with both Elongated and Compact data, we found some evidence that Force and Radial layouts are useful for visualising broad data since users scored them higher with Compact than with Elongated data. Furthermore, we also discovered during our sessions and also from user comments that some participants needed more time to understand Force and Radial layouts. Even though we gave our users explanations and exploration time, our sessions were short to fully understand how these graphs worked.

Moreover, Circle packing achieved slightly higher scores with Elongated data than with Compact data, but it received the lowest scores of all the four layouts in every task overall. This could be due to the unfamiliarity of the users with the layout. It could be again a lack of training in this kind of graph that gave it less advantage. Our studies share some similarities with Zheng and Sadlo[35] that studied hierarchical multivariate visualisations. Especially, they also used Circle packing and mapped their glyphs on this visualisation. Their results aligned with our expectations and their Circle packing also received the lowest scores due to the low readability of the glyphs. Therefore, we suggest that the use of Circle packing could be further restricted to hierarchical data that are not densely crowded, that is trees with very few levels (maximum 3–4), or trees not particularly broad. In the case of broader and deeper hierarchies,[50] analysed the treemap family providing global layouts for balanced and unbalanced trees. Our proposal is aligned with this work since we also try to find the best fitted layout, treemap among others, depending on the nature of the data set.

Conclusion

In this work, we propose a categorisation of hierarchical data into two categories: Elongated and Compact. We then visualise the data using different hierarchical layouts (Tree, Circle packing, Radial, Force) according to that categorisation that can be applied to any hierarchical data.

Our first two research hypotheses propose Tree layout and Circle packing to visualise Elongated structures and Force and Radial layouts to visualise Compact structures . Moreover, we present a formalisation for clustering multivariate data to select the more appropriate way to visualise them, which can be used to cluster any multivariate data. Thus, we designed two glyphs, *one-by-one* and *all-in-one*, and our third hypothesis explores which one is more useful in a global view of data.

We used NewsComTox corpus to present our visualisations. This corpus consists of 4359 comments

posted in response to news articles extracted from online newspapers from August 2017 to July 2020 annotated with toxicity. Then, we conducted a user study with 35 participants to validate our hypotheses. The results indicated that H1 was partially supported as Tree layout (Mean = 4.17 out of 5) with Elongated hierarchies showed significant differences in the score when compared with Force and Radial layouts, but this was not the case with Circle packing. Similarly, H2 was also partially accepted as Force (Mean = 3.69) and Radial (Mean = 3.4) layouts scored significantly higher with compact hierarchies than Circle Packing (Mean = 2.51) but not than the Tree layout. It means that users still preferred the tree layout whenever the structure of the hierarchy is compact. However when we further analysed Force and Radial layouts we discovered that both of them scored higher with Compact structures than with Elongated ones with significant differences. Moreover, we found that *one-by-one* glyph scored higher than *all-in-one* glyph, which aligned with H3.

As ongoing work, we are extending our formalisation to define additional structures such as n-compact and hybrid . We also plan to enhance Force and Radial layouts as from the results we see their potential to visualise Compact data. Moreover, in the future, we intend to study additional glyph designs and perform a user study, including a wider range of user profiles, such as visual design experts, to evaluate them.

Acknowledgements

This work was supported by FairTransNLP-Language: Analysing Toxicity and Stereotypes in Language for Unbiased, Fair and Transparent Systems (PID2021-124361OB-C33) funded by Ministerio de Ciencia e Innovación (Spain). And CI-SUSTAIN: Grant PID2019-104156GB-I00 funded by MCIN/AEI/10.13039/501100011033.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

E Kavaz https://orcid.org/0000-0002-3973-2247
M Nofre https://orcid.org/0000-0001-6883-7657

References

1. Schulz HJ and Schumann H. Visualizing graphs-a generalized view. In: *Tenth international conference on information visualisation (IV'06)*, London, UK, 5–7 July 2006, pp.166–173. New York, NY: IEEE.
2. Hippisley AR, Tariq M and Chang D. Hierarchical data and the derivational relationship between words. In: *Proceedings of the institute for research into cognitive sciences workshop on linguistic databases*, 2001, pp.125–133. Philadelphia, PA: Penn University.
3. Otsuka R, Yano K and Sato N. An organization topographic map for visualizing business hierarchical relationships. In: *2009 IEEE Pacific visualization symposium*, Beijing, China, 20–23 April 2009, pp.25–32. New York, NY: IEEE.
4. NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 2017; 45(D1): D7–D19.
5. Joty S, Carenini G and Ng RT. Topic segmentation and labeling in asynchronous conversations. *J Artif Intell Res* 2013; 47: 521–573.
6. Yang J, Ward MO and Rundensteiner EA. Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. *Comput Graph* 2003; 27(2): 265–283.
7. Qin X, Luo Y, Tang N, et al. Making data visualization more efficient and effective: a survey. *VLDB J* 2020; 29(1): 93–117.
8. Kerren A, Purchase HC and Ward MO. Introduction to multivariate network visualization. In: *Multivariate Network Visualization*. Cham: Springer, 2014, pp.1–9.
9. Schulz HJ. Treevis.net: A tree visualization reference. *IEEE Comput Graph Appl* 2011; 31(6): 11–15.
10. Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In: Kaufmann M (ed.) *The Craft of Information Visualization*. Amsterdam: Elsevier, 2003, pp.364–371.
11. Ward MO. Multivariate data glyphs: Principles and practice. In: *Handbook of Data Visualization*. Berlin: Springer, 2008, pp.179–198.
12. Nobre C, Meyer M, Streit M, et al. The state of the art in visualizing multivariate networks. *Comput Graph Forum* 2019; 38: 807–832.
13. Schulz HJ, Hadlak S and Schumann H. The design space of implicit hierarchy visualization: a survey. *IEEE Trans Vis Comput Graph* 2011; 17(4): 393–411.
14. Meeks E. *D3. Js in Action: Data Visualization with JavaScript*. New York, NY: Simon and Schuster, 2017.
15. Darzi Y, Yamate Y and Yamada T. Functree2: an interactive radial tree for functional hierarchies and omics data visualization. *Bioinformatics* 2019; 35(21): 4519–4521.
16. Archambault D, Munzner T and Auber D. Grouseflocks *steerable exploration of graph hierarchy space*. *IEEE Trans Vis Comput Graph* 2008; 14(4): 900–913.
17. Hoque E and Carenini G. Convis: a visual text analytic system for exploring blog conversations. *Comput Graph Forum* 2014; 33: 221–230.
18. Hu CC, Wei HX and Chi MT. Shareflow: A visualization tool for information diffusion in social media. In: *International conference on ubiquitous information management and communication*. Cham: Springer, pp.563–581.
19. McNutt AM and Kindlmann GL. Improving the scalability of interactive visualization systems for exploring threaded

conversations. In: *EuroVis (Posters)*. Portugal: The Eurographics Association, pp.53–55.
20. Burch M, Vramulet A, Thieme A, et al. Vizwick: a multiperspective view of hierarchical data. In: *Proceedings of the 13th international symposium on visual information communication and interaction*. New York: Association for Computing Machinery, 2020, pp.1–5.
21. Kerren A, Purchase H and Ward MO. *Multivariate Network Visualization (proc. dagstuhl seminar 13201)*. Switzerland: Springer, 2014.
22. Keim DA. Visual techniques for exploring databases. In: *International conference on knowledge discovery in databases (KDD'97)*, Newport Beach, CA, USA, November 1997.
23. Keim DA and Kriegel HP. Visualization techniques for mining large databases: A comparison. *IEEE Trans Knowl Data Eng* 1996; 8(6): 923–938.
24. Wong PC and Bergeron RD. 30 years of multidimensional multivariate visualization. *Scientific Visualization* 1994; 2: 3–33.
25. Bezerianos A, Chevalier F, Dragicevic P, et al. Graphdice: A system for exploring multivariate social networks. *Comput Graph Forum* 2010; 29: 863–872.
26. Cao N, Sun J, Lin YR, et al. Facetatlas: multifaceted visualization for rich text corpora. *IEEE Trans Vis Comput Graph* 2010; 16(6): 1172–1181.
27. El-Assady M, Sevastjanova R, Gipp B, et al. NErex: named-entity relationship exploration in multi-party conversations. *Comput Graph Forum* 2017; 36: 213–225.
28. Borgo R, Kehrer J, Chung DH, et al. Glyph-based visualization: foundations, design guidelines, techniques and applications. In: *Eurographics (state of the art reports)*, Girona: The Eurographics Association 2013, pp.39–63.
29. Du F, Plaisant C, Spring N, et al. Visual interfaces for recommendation systems: finding similar and dissimilar peers. *ACM Trans Intell Syst Technol* 2019; 10(1): 1–23.
30. Xu J, Tao Y, Lin H, et al. Exploring controversy via sentiment divergences of aspects in reviews. In: *2017 IEEE pacific visualization symposium (PacificVis), Seoul*, 18–21 April 2017, pp.240–249. New York, NY: IEEE.
31. Sun G, Tang T, Peng TQ, et al. Socialwave: visual analysis of spatio-temporal diffusion of information on social media. *ACM Trans Intell Syst Technol* 2018; 9(2): 1–23.
32. El-Assady M, Gold V, Acevedo C, et al. Contovi: Multiparty conversation exploration using topic-space views. *Comput Graph Forum* 2016; 35: 431–440.
33. Jarvenpaa SL. The effect of task demands and graphical format on information processing strategies. *Manage Sci* 1989; 35(3): 285–303.
34. Carpendale MST. *Considering Visual Variables as a Basis for Information Visualisation*. Calgary: University of Calgary, 2003.
35. Zheng B and Sadlo F. On the visualization of hierarchical multivariate data. In: *2021 IEEE 14th pacific visualization symposium (PacificVis)*, Tianjin, China, 19–21 April 2021, pp.136–145. New York, NY: IEEE.
36. Gortler J, Schulz C, Weiskopf D, et al. Bubble treemaps for uncertainty visualization. *IEEE Trans Vis Comput Graph* 2018; 24(1): 719–728.
37. O'Handley B, Wu Y, Duan H, et al. Treevisual: design and evaluation of a web-based visualization tool for teaching and learning tree visualization. In: *Proceedings of American society for engineering education annual conference*. 2022.
38. Stasko J and Zhang E. Focus + context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In: *Proceedings of the IEEE Symposium on Information Visualization 2000 (INFOVIS 2000)*, 2000, pp.57–65. New York, NY: IEEE.
39. Kruskal JB and Landwehr JM. Icicle plots: better displays for hierarchical clustering. *Am Stat* 1983; 37(2): 162–168.
40. Von Landesberger T, Kuijper A, Schreck T, et al. Visual analysis of large graphs: state-of-the-art and future research challenges. *Comput Graph Forum* 2011; 30: 1719–1749.
41. Wagemans J, Elder JH, Kubovy M, et al. A century of gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychol Bull* 2012; 138(6): 1172–1217.
42. Cheong SH and Si YW. Force-directed algorithms for schematic drawings and placement: a survey. *Inf Vis* 2020; 19(1): 65–91.
43. Taulé M, Ariza A, Nofre M, et al. Overview of detoxis at iberlef 2021: detection of toxicity in comments in spanish. *Procesamiento del Lenguaje Natural* 2021; 67: 209–221.
44. Kavaz E, Puig A, Rodriguez I, et al. Data visualization for supporting linguists in the analysis of toxic messages. In: *WSCG 2021, 29th international conference in central Europe on computer graphics, visualization and computer vision held in co-operation with the EUROGRAPHICS Association*, 17–20 May 2021.
45. Brunnström K and Barkowsky M. Statistical quality of experience analysis: on planning the sample size and statistical significance testing. *J Electron Imaging* 2018; 27: 1–11.
46. Anjard RP. Management and planning tools. *Train Qual* 1995; 3: 34–37.
47. Manning A and Amare N. Visual design principles and effective sentence diagrams for the 21st century. In: *2014 IEEE international professional communication conference (IPCC)*, Pittsburgh, PA, USA, 13–15 October 2014, pp.1–8. New York, NY: IEEE.
48. Borges J. VisAC: an interactive tool for visual analysis of consanguinity in the ancestry of individuals. *Inf Vis* 2022; 21(4): 354–370.
49. Hoffswell J, Li W and Liu Z. Techniques for flexible responsive visualization design. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York: Association for Computing Machinery, 2020, pp.1–13.
50. Schreck T, Keim D and Mansmann F. Regular treemap layouts for visual analysis of hierarchical data. In: *Proceedings of the 22nd spring conference on computer graphics (SCCG '06)*, 2006, pp.183–190. New York, NY: Association for Computing Machinery.