HyperStorylines: Interactively untangling dynamic hypergraphs

Vanessa Pena Araya, Tong Xue, Emmanuel Pietriga, Laurent Amsaleg, Anastasia Bezerianos

HyperStorylines: Interactively Untangling Dynamic Hypergraphs

Journal Title
XX(X):1–21

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Vanessa Peña-Araya[1], Tong Xue[1], Emmanuel Pietriga[1], Laurent Amsaleg[2], and Anastasia Bezerianos[1]

Abstract

We present the design and evaluation of HyperStorylines, a technique that generalizes Storylines to visualize the evolution of relationships involving multiple types of entities such as, *e.g.,* people, locations and companies. Datasets which describe such multi-entity relationships are often modeled as hypergraphs, that can be difficult to visualize, especially when these relationships evolve over time. HyperStorylines builds upon Storylines, enabling the aggregation and nesting of these dynamic, multi-entity relationships. We report on the design process of HyperStorylines, which was informed by discussions and workshops with data journalists; and on the results of a comparative study in which participants had to answer questions inspired by the tasks that journalists typically perform with such data. We observe that although HyperStorylines takes some practice to master, it performs better for identifying and characterizing relationships than the selected baseline visualization (PAOHVis) and was preferred overall.

Keywords

hypergraphs, data journalism, storylines, comparative study

Introduction

Storyline visualizations communicate the evolution of relationships between different entities over time. Entities are represented by curved lines, that come close together to indicate the start of a relationship between them, and drift apart when that relationship ends. Introduced in an XKCD comic[1] in 2009, storyline visualizations have since been used to effectively represent narratives of relationships in different domains, such as software engineering[2], genealogy[3], social network analysis[4], and literary studies[5].

This simple yet powerful representation effectively conveys the notion of a *story* about how those relationships evolve over time. But it typically assumes one type of entity and one type of relationship only, which corresponds to fairly simple dynamic graphs with one type of vertex and one type of edge. For instance, in storylines of movies, curved lines represent characters, that come together to indicate the spatio-temporal co-occurence of two or more of them. Characters are the graph's vertices, and co-occurences its edges. Both only exist in the relevant intervals of the dynamic network's timeline.

Many datasets have a more complex structure, however. Multivariate networks often involve multiple types of vertices and edges. In some cases, each edge can relate more than two vertices, forming *hypergraphs*[6]. Adding further complexity, these *hyperedges* can relate vertices of different types. Let us consider data journalism and politics in particular – the application area that motivated our work. A storyline visualization can be drawn based on the different elections that a country's prominent politicians have been involved in over the years. In this case, lines coming closer together in some time-span could indicate politicians running for the same elected office. But the dataset can actually contain much more: where and when political meetings have been held, prior offices held, affiliation to political parties, connections with companies and social connections (*e.g.,* common colleagues). These represent multiple additional types of vertices and edges. The resulting hypergraphs can be quite tangled and difficult to represent.

In the considered application area – as in several others – understanding the temporal aspect is key to the data analysis. We thus take the concept of Storyline visualizations, that works well for representing simple dynamic relationships, and push it further. We introduce *HyperStorylines*, an extension of the storyline approach to *dynamic hypergraphs*. Contrary to classic storylines (one type of entity, one type of relationship), with HyperStorylines users can get details on-demand about other entities and their relationships through a combination of (i) *nesting*, (ii) *aggregation* and (iii) *reconfigurable views*.

We start from a representation similar to a classic storyline (Figure 1-A), showing how one type of entity (people) relate along a second type of entity (time). Users can get details on-demand about particular relationships. These details come in the form of a more detailed storyline *nested* in the main one, typically involving other types of entities. Figure 1-B shows such a nested storyline, obtained by clicking on the relationship highlighted in red in Figure 1-A. The nested

[1]Université Paris-Saclay, CNRS, Inria, LISN
[2]Inria, Univ Rennes, CNRS, IRISA

Corresponding author:
Vanessa Peña-Araya Université Paris-Saclay, CNRS, Inria, LISN. Université Paris Saclay - Bât. 660, 91405 ORSAY Cedex, FRANCE
Email: vanessa.pena-araya@inria.fr

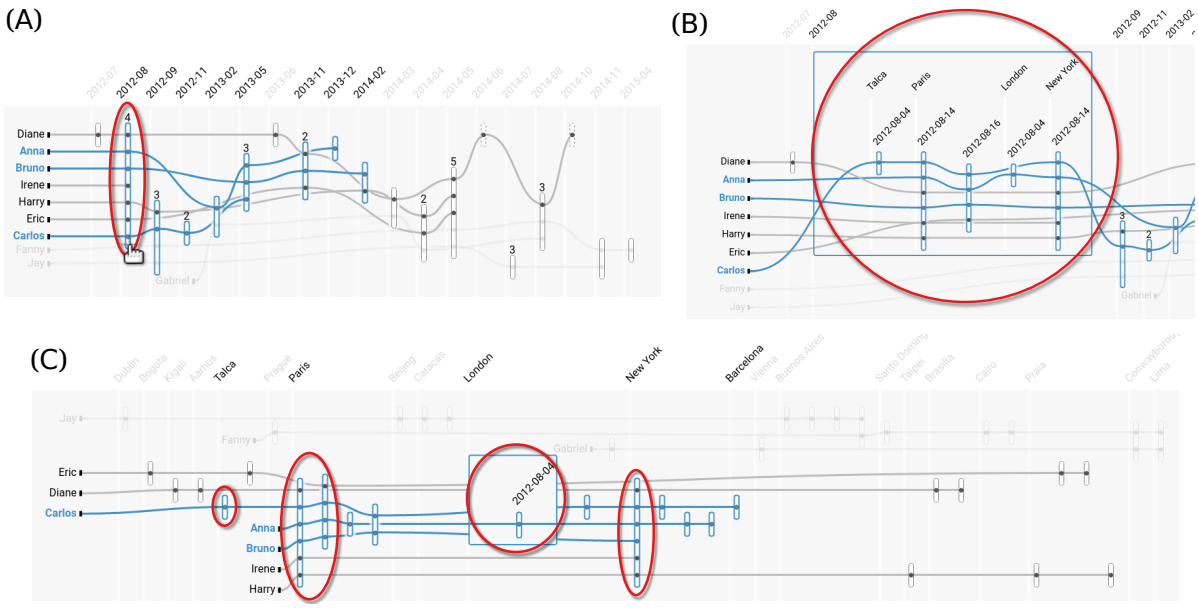

Figure 1. (A) Lines are the stories of entities of one type, in this case of the type *people*, that evolve along the horizontal axis, that here represents entities of a second type which is *time* (aggregated by months). Small vertical bars are constructed relationships, positioned in the intersection of both axes of the entities that compose them. In our case, these relationships represent entities that appeared in news articles. For example the relationship highlighted in red indicates all people that appeared in news articles in August 2012. Relationships can have zero or more internal **nested** entities (a third type of entity). These nested entities can be seen as a mini-story by interactively expanding the relationships (B). For example, here we see details in August 2012, including the nested *location* entities that tie people to places in the articles, and more precise date information. The type of entities on the horizontal axis, the vertical axis and the nested entities can be changed with selectors in our tool. For example, in (C) we can see the stories of people related by locations instead of time (time is the nested entity). The red circles across images indicate where the entities that contribute to the highlighted relationship in (A) appear in the other views.

storyline in Figure 1-B gives information about the locations visited by multiple people in a given time period.

Storyline nesting works in conjunction with an *aggregation* mechanism to make navigation in the data more scalable. In our example, aggregation is illustrated on the temporal dimension (visits are displayed by day rather than month in Figure 1-B). HyperStorylines also lets users interactively *reconfigure* the view to explore different aspects of the hypergraph. Back to the same example, journalists might want to explore spatio-temporal co-occurences (participation to the same event), affiliation history or election campaign history – which are all different types of relationships. Figure 1-C illustrates such a change of perspective, which helps journalists understand how different people relate to specific locations regardless of time.

Our approach can: (i) express complex and diverse relationships that include multiple types of entities, by providing a main storyline view (context) and details on demand through nested entities within a relationship (focus); (ii) scale to a large number of entities, by seeing them at different granularity levels in the main story, or in the nested entities within a relationship; (iii) generalize traditional storylines by extending the concept of a "story" to express relationships across different dimensions beyond time.

The design of HyperStorylines was informed by discussions and workshops with data journalists analyzing complex relationships between political figures in regional elections in France, and their roles within local councils, political parties, organizations and in connection to news events. We compared HyperStoryline with PAOHvis[7], one of the most recent tools for visualizing dynamic hypergraphs and the most appropriate baseline for our context, using a set of tasks identified based on our discussions with the journalists. We found that HyperStorylines better supported tasks that required identifying and characterizing relationships involving several entities. However, participants performed better with PAOHvis when searching for large relationships, *i.e.*, those with a large number of entities involved.

Related Work

Visualization research is often motivated by data journalism, from the seminal work of Segel and Heer[8] on narrative visualizations and their structure, to studies using visualizations from news articles as examples to investigate specific questions, such as the influence of titles in visualization reading[9,10]. The notion of a narrative or story is key in these previous works and in the domain of data storytelling[11]. Nevertheless, when we refer to stories in our work, we mean the evolution of entities (people, organizations, locations) in a dataset, in a way similar to past work on storylines that focus on the evolution of people.

While we are motivated by data journalism, our goal is not to create polished visualizations that tell one news story about the data, but rather to provide an interactive visualization to help domain experts working with dynamic hypergraphs (including data journalists) to explore their datasets. In that respect, our goal is rather aligned with that of Brehmer *et al.*[12]. Their design study, and the resulting Organizer system, focus on helping journalists explore large collections of documents. With HyperStorylines, we focus

on visualizing the evolution of specific entities and their inter-relationships, rather than the documents themselves.

Next, we give an overview of the literature about visualizing dynamic hypergraphs, storyline visualizations, and related empirical studies.

Visualization of Dynamic Hypergraphs

Similar to graphs, *static* hypergraph visualizations include node-link diagrams [13–15], matrices [16,17], or combinations of visualizations in custom applications (*e.g.,* a query based exploration interface relying on sorted histograms and tabular visualizations [18]). Representing hypergraphs using node-link diagrams is mainly done by adding an extra type of node that represents a hyperedge, which is connected to multiple vertices/entities [13–15]. Adding such elements increases the visualization's complexity, a problem that can be alleviated by using appropriate layout strategies [19]. Representing hypergraphs using matrices is mainly done by creating incident matrices, where different entity types are added together as columns or rows of the matrix. Their type can be differentiated by their position, their name or a color given to their name [16]. With the exception of HYPER-MATRIX [17] (discussed later), these matrix representations do not typically consider evolution over time.

Dynamic Graphs and Hypergraphs. A recent survey by Beck *et al.* [20] identifies three main dynamic graph visualization strategies: mapping the time dimension to an animation [21,22], mapping time to a spatial dimension representing a static timeline [23–25], and combinations of both [26]. Animation can be an appealing way to represent change over time but it may be harder to perceive and conceptualize [27]. TimeArcTrees [23] and an extended design of Storylines by Arendt and Blaha [25] both fall in the second category of static timelines. They use horizontal lines to represent entities and arcs to represent relationships between two entities. However, they quickly becomes illegible as the number of connections among entities increase.

To the best of our knowledge, PAOHvis [7] and HYPER-MATRIX [17] are the two approaches that can visualize large hypergraphs without generating considerable clutter. In PAOHvis, vertices are represented by rows. Hyperedges are vertical lines ordered by time, linking all involved entities by marking them with dots. Different types of entities are assigned a specific color. Although PAOHvis displays hyperedges in an organized and uncluttered way, it can be hard to follow all the entities involved in a relationship when there is a large number of them. HYPER-MATRIX uses a matrix-based representation, with several levels of semantic zoom representing different levels of granularity. Time is represented by a glyph in each cell, which can make the analysis and comparison of relationships (hyperedges) over time a complex task.

With few exceptions, visualizations of graphs and dynamic graphs are not well suited to hypergraphs. Additionally, the two visualizations specifically designed for dynamic hypergraphs have some drawbacks when it comes to representing relationships between multiple entities of different types. HyperStorylines aims to provide a less cluttered visualization of hypergraphs, allowing users to unravel relationships progressively and to follow individual entities or links between entities.

Storylines Visualizations

Storylines visualizations became popular after Randall Munroe published several hand-drawn narrative movie charts on his XKCD comics website [1]. In these charts, each character is a horizontal line whose length represents the character's lifespan in the movie. Lines converge to meet other lines when the characters they represent interact with each other in the movie.

The first known software that automatized the creation of the XKCD narrative charts was introduced by Vadim Ogievetsky [28]. The same year, Ogawa and Ma [2] published the first algorithm to create automatic storylines visualizations. Follow-up research about storylines has mainly focused on improving their layout [29–32], which is not the focus of our work. Taking a different approach, Tang *et al.* [33] conducted two user studies to understand how people manually create storylines. This in turn informed a design space that maps narrative elements to visual elements. Based on these, they developed iStoryline, a tool that allows users to interactively enhance an automatically generated storyline narrative. Similarly, PlotThread [34] is a tool that uses a reinforcement learning model to assist users in the creation of well-optimized storylines visualizations.

Besides the layout of entities, additional visual encodings are necessary in storylines to represent other contextual information about the narrative. Common approaches are annotations to give location names [29,30] and color to show similar entities [1]. But these do not scale well to multiple locations or entity types, and do not allow to effectively follow contextual information over time. Another approach was introduced by Arendt & Pirrung [35], in which the y-coordinate indicates the context in which interaction between characters takes place (*e.g.,* locations such as a cafe or a theater are at a specific y-position). The practicality of this approach is unclear, however, in cases where more than one context is active at one time step.

HyperStorylines aims to provide visualizations with the potential to encode more information, that scale to a larger number of entities, while causing less clutter than traditional storylines. It also aims to support the visualization of relationships along dimensions other than time, while keeping the same visual metaphor.

Related Empirical Studies

Empirical studies about graph visualization readability abound. Some evaluate different representation strategies [36], while others focus on very specific aspects such as edge curvature [37]. Those that consider dynamic graphs can be broadly categorized as follows: those studying the effect of consistent layout, those comparing animation and small multiples, and those focused on specific applications [20].

We only found few studies that evaluate visualizations of hypergraphs or dynamic hypergraphs. These few either do not conduct a controlled user study [17,19,38] or they do not compare their tool with other visualizations [7]. There is thus a lack of studies comparing hypergraph techniques.

Studies on Storylines visualizations are focused on the layout[30,33,35], with two exceptions. The work of Arendt & Pirrung[35] studies the effect of using the y-coordinate to represent contextual information, but does not compare the approach to other hypergraph visualizations. Zhao *et al.*[39] compare their storylines-based visualization with two traditional representations of dynamic graphs, but do not consider multiple types of entities (hypergraphs).

Our study is the first to compare a hypergraph visualization using a storyline approach, to a competitive alternative specifically designed for dynamic hypergraphs.

Workshops with Journalists: Relationship Inquiries in Investigative Journalism

While HyperStorylines can accommodate any hypergraph that includes temporal information, the technique was originally motivated by the needs of investigative journalists looking for possible connections between political figures and organizations. In 2018-2019, we conducted two workshops with staff from Ouest France[40], the most read francophone newspaper in the world with 2.5 million daily readers. Details on the roles and experience of our expert participants can be found in Table 1.

Participant ID	W#	Role	Experience
J1	W1 & W2	Data Journalist	21 yrs
C1	W1	Data Curator & Editor	13 yrs
S1	W1	Support Staff	10 yrs
S2	W1	Support Staff	6 yrs
J2	W2	Data Journalist	15 yrs
S3	W2	Support Staff	22 yrs
S4	W2	Support Staff	5 yrs

Table 1. Information on participants that took part in the two workshops (W1,W2) - only J1 was present in both.

In the first half-day workshop, we met with a data journalist, an information curator managing newspaper archives, and two support staff with journalistic training who assist journalists with information search and fact checking. The goal of this first workshop was to understand the challenges journalists had faced in the preparation of recent news stories, and to identify the types of questions they had when investigating possible relationships between political entities. The workshop started with the research team presenting their expertise. The journalists and staff then presented the tools they use, which we discuss next, and the process they follow to construct news articles. To motivate and inspire discussion, the research team introduced existing visualization tools to visualize relationships extracted from document collections, including Jigsaw[41] and TimeArcs[23] (PAOHvis[7] was not yet published).

In the second half-day workshop, we met with two journalists (one present in the previous workshop) and two support staff. The goal of this 2nd workshop was twofold: to validate and verify the tasks and challenges identified in the first workshop, and to get early feedback on a first version of our prototype visualization. The workshop started with the research team reminding participants about the tasks that had emerged from the first workshop, seeking confirmation and new concrete examples. It continued with a demo of an early version of our tool.

We now report on findings from the workshop. We quote the participants' ID when applicable. However, we note that as this is a participatory workshop, participants often built upon each-others' comments. We thus list all participants who initiated the discussion (usually one of the journalists or the data curator) but note that all participants confirmed and/or contributed to the findings.

The journalistic staff have at their disposal a large number of text document collections in the form of electronic and digitized paper archives of news articles over several decades. They currently look at this information using an in-house search engine (Troove) that provides faceted browsing of their digitized document collection using keywords, political entities, years, *etc.* All participants, in both workshops, indicated they used this tool extensively, but that they also sometimes relied on external sources such as archives of other news networks and official government repositories, confirming information with associates in political offices or in the government. The two data journalists (J1,J2) often use tools like OpenRefine to clean data, and have created over the years large Excel sheets with tabulated information that connects interesting facts back to specific articles in archives that they revisit and update.

Part of the work of their IT department is to conduct *entity extraction* from these documents (Named Entity Recognition - NER) in order to help journalists: (i) search for specific people, organizations, political parties and other entities; and (ii) get statistics on frequently mentioned entities over time. Their current search tool helps them see information about a specific entity, such as a particular politician, but does not support understanding the interconnections between multiple entities. So beyond the tasks that focus on one entity at a time, our experts (J1, later confirmed by J2) explained how their questions and needs are often centered around understanding how multiple entities of interest are interconnected and how their relationships evolve, a task not well supported by their current tools.

Q1: Understand the temporal inter-relationships between multiple entities. There are several entities of interest in investigative journalism, such as people, companies, political parties, councils, social movements, *etc.* The data journalists (J1,J2) and support staff (C1,S1) explained that they often investigate the trajectory of such entities and their relationships in depth. As an example, one journalist (J1) mentioned the preparation of a story reviewing the evolution of the French social movement *Gilets jaunes*, that at the time encompassed 11,000 online news articles. In particular they wanted to track when the movement appeared in the news, how coverage increased, and most importantly whether the press associated them with other entities such as political figures (*e.g.,* president Emmanuel Macron) or specific topics (*e.g.,*, social debate, pensions, demonstrations, material damage), and how these relationships evolved over time. Currently, the journalists construct these timelines by searching for the names of entities and related keywords (J1,J2), and then look over the articles to understand how they relate to each other and how their connections evolve over time. They often need to double-check these relationships (S1,S2), sometimes using multiple resources - as one journalist mentioned: *"I try to confirm a relationship from at least two sources"* (J2).

Q2: Gather information about different types of entities characterizing a relationship. The example of specific topics and political figures mentioned above includes two types of entities. Our journalists explained that relationships are often more complex (J1,J2) and can involve several types of entities, such as places, organizations, *etc.* They gave the example (J2,S4) of a political candidate who appeared quite suddenly ("seemingly out of nowhere") and gained a lot of popularity and campaign support. The journalist and stuff wanted to find out if the candidate had connections with other political or public figures, which would have been previously overlooked. As they explained, they were not just interested in possible relationships with others, but also the nature of these relationships (*e.g.,* served in the same political party, were associated with the same company even if not at the same period, held similar positions in different organizations during the same time period). This also led journalists to mention an imaginary scenario (J1) of investigating two companies and finding out that their connection is a common board member. As they explained, being able to see different facets or properties of how entities are connected is important. Currently, journalists need to identify relationships across articles (*e.g.,* news articles that mention both figures) and dig for the details in the article keywords and the raw text - a process described by both journalists and all staff.

Q3: Similar paths / evolutions. The data journalists expressed how they sometimes want to compare and identify similarities in the evolution of political figures (J1,J2,S4), as this could help them predict career paths or may simply make an interesting story. When asked what would constitute a "similar" path, they explained that this would depend on the figure considered. For example, for contemporary politicians, "similar" could mean that they may have been connected with the same people (J1), or that they were politically active in similar time periods (J2). For political figures across the years, "similar" could mean that they were involved in the same councils or organizations (J1,S4), or that their activities had followed similar patterns, *e.g.,* periodic bursts of making the news that are close in time (J2), or increase of political activity once they started a new role (J1). Currently, journalists and stuff need to search for each figure and construct their timeline by hand.

Q4: Massive events. In both workshops, the two data journalists (J1,J2) and the curator (C1) stressed the importance of being able to identify massive events, *i.e.*, events that involve many people. They gave us collectively examples related to the recent regional elections, where several city councils (that may include 20 or more people) stepped down en masse to protest against newly introduced government policies. The journalists explained that they would like to identify and investigate these events in more depth, as well as if-and-how they may be related. For example, if they follow closely each other in time (a chain reaction); if they are regional (neighboring councils); if they have possible instigators (members that have moved across councils and have been involved in such mass events multiple times), *etc.* It is currently hard to identify such massive events, other than browsing for articles that have many entities extracted from them.

Based on these questions we saw an opportunity to support journalists in exploring relationships across multiple entities. The journalistic staff provided us with the entity extractions and original articles they came from, that we then used to generate a hypergraph that forms the basis of the dataset used in our evaluation.

The questions and needs mentioned above were identified in the first workshop and confirmed in the second. The second workshop also gave journalists and support staff the opportunity to view early versions of our designs. We report on their feedback related to our design later in the paper (see Section Feedback).

Previous work on data journalism[12] has focused on helping journalists explore large document collections (visual document mining). Our work is orthogonal - the scenarios and questions described by our journalists and staff are generally focused on specific entities, that may be small in number, and their inter-relationship. It thus attempts to give overview and details of these relationships, rather than the documents themselves. Nevertheless, we can see cases where the two approaches could be combined, with the Overview system[12] allowing journalists to identify topics or people of interest, and HyperStorylines to explore their interconnections. The questions identified in this section follow for the most part the second general pattern (task abstraction) identified in the Overview system[12] work: starting from a hypothesis (suspected connections between entities) and looking for evidence and details.

HyperStorylines

We now describe the HyperStorylines* technique in detail, using the same dataset throughout, which was provided by Ouest France and forms the basis of our motivating scenario. The dataset is composed of a set of documents from archived news articles that contain named entities of type people, locations and organizations (see Figures 1-4). This kind of dataset can be modeled as a hypergraph, where each news article is modeled as a hyperedge linking all the named entities mentioned in it. In other words, each entity is a node (of any of the above types); and the relationships between them given in the news articles are hyperedges (connecting two or more nodes of any type). In our case, we also consider the date when the article was published as an entity. Articles published on the same date and containing the exact same entities are merged as one hyperedge, storing the reference to all the articles that contain that set of entities. All data for Figure 1 and the figures of this section have been anonymized as per a confidentiality agreement with Ouest France.

Nesting Hyperedges in Storylines

Trying to show all types of entities at the same time in a conventional storylines visualization would be highly confusing in most cases, and would result in storylines that mix multiple types of relationships (*i.e.*, hyperedges connecting varying types of nodes). Instead, regardless of the number of entity types that a hypergraph may contain, we choose to only show two main types (*e.g.,* people

*https://gitlab.inria.fr/ilda/hyperstorylines

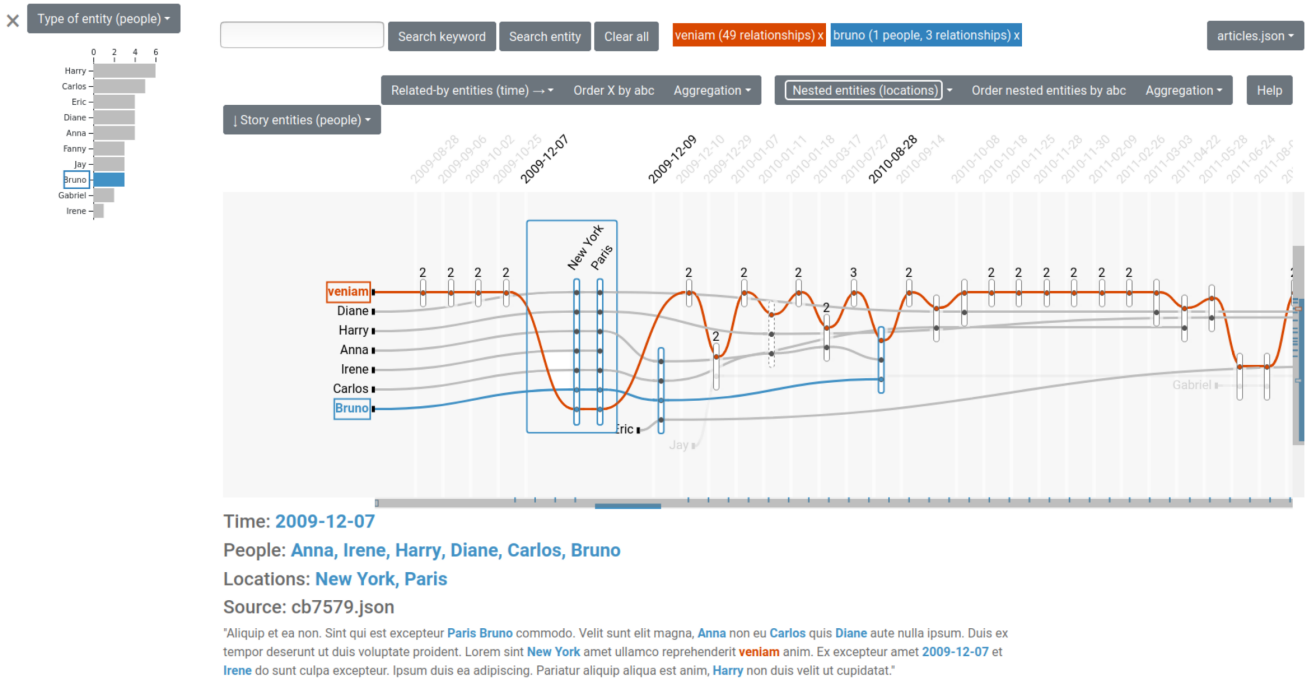

Figure 2. Prototype VA system for our data journalism scenario, based on a HyperStorylines visualization (central panel). Left panel: an overview histogram of the number of hyperedges per node - the node type can be changed with the selector at the top. Top panel: search field that allows users to search either for a node by name (in blue); or for a text term in the original news articles (in orange) that is shown as a special search entity. When interactively expanding a hyperedge, the bottom panel shows provenance information about this hyperedge (original article from which it was derived), including a summary of the named entities involved.

and locations) simultaneously and the corresponding part of relevant hyperedges. Users can select the two types of entities to start their exploration with, and then selectively unroll some hyperedges to reveal their full complexity progressively, which reveals the nodes' relationships with other types of entities in the dataset.

Figure 1-A and Figure 1-C illustrate two different configurations to start from: *people* over *time* in the former case, *people* by *location* in the latter case. Starting from Figure 1-A, a journalist might be interested in the connection between Anna, Bruno and Carlos in August 2012. This connection bar (circled in red in the figure) represents a hyperedge derived from one or more news articles which were published during that month. The journalist can unroll this particular bar to reveal (Figure 1-B) additional nodes that the hyperedge connects to, in this case those of type *location*. The resulting *nested* storyline shows how all three people (as well as others) connect to Paris on both 2012-08-14 and 2012-08-16, and with New York on 2012-08-14. Thus a view can simultaneously show three types of entities (two main types and one nested), but users can interactively switch between them to access other types, as discussed later.

Interacting with HyperStorylines

HyperStorylines are highly interactive visualizations, that will typically be embedded in a larger visual analytics system. We describe one such system, whose design is informed by our data journalism use case. It is composed of four main components, illustrated in Figure 2.

In this section we describe how these components work together to assist analysts interactively explore dynamic hypergraphs whose hyperedges connect nodes of different types. We call *relationship bar* the visual representation of hyperedges (or subparts thereof, depending on whether they have been fully unrolled or not).

Constructing views. HyperStorylines allows analysts to progressively construct partial views of the dataset, focusing on specific entities. These partial views are constructed and displayed as the main visualization at the center of the interface. Users can select which types of nodes to put on the horizontal and vertical axes. The system then generates a HyperStorylines visualization with hyperedges involving nodes of those types.

As in traditional storylines visualizations, each curved line represents the story of one node (*e.g.,* each individual in Figure 1) that progresses along the horizontal axis. We call the nodes that have this role in a particular view configuration *story entities*. Nodes on the horizontal axis, that define the criterion according to which story entities get connected, are called *related-by entities*. For example, in Figure 3-A, the story entities are of type *people*, and are related-by entities of type *time*. Story entities, each one a line, bend and come together to form constructed relationships with each other, according to the corresponding related-by entity (*e.g.,* Anna, Bruno, Carlos and other people in 2012-08-14). Constructed relationships are represented as vertical bars through which the story entity lines pass through. Such relationships, with time as a related-by entity, are central for the sort of analysis that our users perform (Section Workshops, *Q.1*).

By selecting which type of entity will act as story entities and which as related-by entities, users can construct a view that involves any combination of two types of entities. This generalization allows users to decide what is important at different stages of the analysis. For instance, after observing

the evolution of relationships between people over time in Figure 3-A, we might be interested in seeing how people could be related-by locations or local councils (Section Workshops, *Q.3*). Such a view can be seen in Figure 3-B, where people remain the story entities but now locations play the role of related-by entities. In other words, this view represents the stories of people over places (instead of over time). This view allows users to answer, for example, questions such as *"How many people were involved with a specific location?"*, or *"Which groups of people may meet in the same locations?"*.

We note that a hyperedge in the dataset is not represented in the same manner in all views of HyperStorylines. This representation will depend on which nodes they connect and what roles the node types play in the current view configuration (story entities, related-by entities, or neither of these roles). Going back to Figure 1, the red annotations show how one hyperedge represented as a single bar in Figure 1-A gets split in multiple bars in Figure 1-C, as the node type playing the role of related-by entities has changed.

This design decision was made after iterating with the journalists. We had originally designed HyperStorylines to allow for multiple node types to act as story entities (*e.g.,* people and locations were stories represented by lines, and they formed relationships by time). But feedback from journalists indicated that such views were somewhat overwhelming (see Section Feedback for details). Their focus (Section Workshops, *Q.2*) is most often on relationships between one type of entity first (people). Only then do they gather information about characteristics of this relationship (connected in time through locations, organizations, *etc.*).

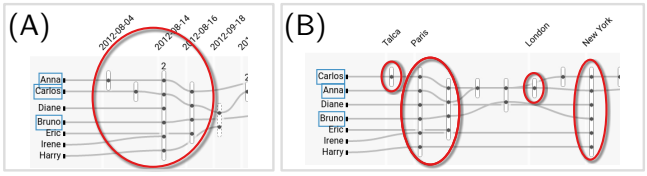

Figure 3. Two constructed views with the three people highlighted in Figure 1. (A) People related-by time without temporal aggregation. (B) People related-by locations. Encircled in red are the same hyperedges marked in Figure 1.

Creating aggregated relationships. By default, the lines displayed in a constructed view will be at the same granularity level as the hyperedge in the dataset. For example our dataset hyperedges come from co-occurrences in news articles, that are published on a specific day. This might be too detailed, however, and users have the possibility to aggregate by day, month or year when dealing with time on the horizontal axis. Figure 3-A shows hyperedges without temporal aggregation. Figure 4-A shows the same hyperedges aggregated by month. These levels of aggregation allow users to see higher-level patterns. For example, we can see that there are news articles about Anna and Carlos in August 2012 but not between September 2012 and February 2013. By default, HyperStorylines allows users to aggregate either based on time or geography (using administrative levels). However, users can extend this by creating custom aggregation classes depending on their own data, as detailed in the software supplemental material.

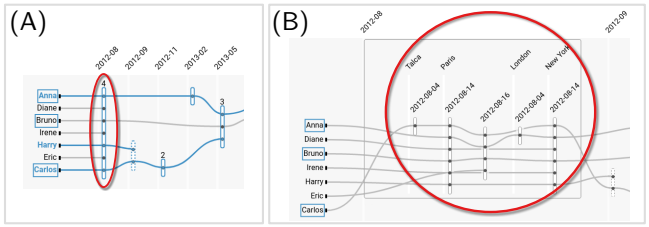

Figure 4. Two constructed views with the three people highlighted in Figure 1. (A) People related-by time aggregated by month. (B) Expanded hyperedge to show locations involved without temporal aggregation. Encircled in red is the same hyperedge marked in Figure 1.

Interactively unfolding other types of entities. Relationships between people over time or over locations are often not enough, as journalists may want to see exactly how these people are connected (*e.g.,* by organization, location, *etc.*, see Section Workshops, *Q.2*). Once users find an interesting relationship, they can get details about the other entities involved that were hidden from the view they constructed. These entities are nested inside relationships. We call them *nested entities*. These entities are only visible when interactively expanding a relationship. Within the constrained space of the expanded relationship, nested entities replace the related-by entities, creating a mini-storyline visualization of story entities grouped by the nested entities. The type of nested entities can be selected with a widget in the interface. For example, in Figure 4-A, we see that story entities are people, related-by time. When we expand the constructed relationship in 2012-08 (Figure 4-B), we see the connection of these people at locations Talca, Paris, London and New York. We also see time with another aggregation level.

The number of nested entities inside a relationship is given by a number above it. If a relationship has **no nested entities** (*i.e.*, it does not include entities of the type given by the selector 'inner entities'), the relationship has dashed border. For example, the hyperedge in September 2012 in Figure 4-A indicates that Harry and Carlos were mentioned on that date but they were not associated with a location. **Only one nested entity** is indicated by a solid border and no number above it, like the hyperedge involving Anna in Figure 4-A at date 2013-02. **More than one nested entity** is indicated by a solid border and the number of nested entities above its relationship bar. This is the case of the tallest constructed relationship in Figure 4-A.

When expanding a hyperedge, the original article(s) from where it was extracted will appear at the bottom of the user interface (Figure 2).

Creating Search Entities and other interactions. HyperStorylines also allows users to create new types of relationships that show the intersection of a text term with the entities already shown in a view. This can be done by searching for a term or keyword using the search bar at the top. The search term becomes a new type of entity, a **search entity**, and creates a set of **search relationships** between it and the entities appearing in the articles that contain that term. For the moment, the search is based in textual co-occurrences of

a term in the whole news article from where relationships were extracted. However, as the search is implemented using ElasticSearch, it would be possible to implement more complex queries using, *e.g.,* logical operators (`AND`, `OR`, `NOT`). Each keyword searched will appear as another story line in the main view, intersecting all the hyperedges that were extracted from an article that contains that keyword. These special entities representing text searches are marked by an orange rectangle to differentiate them from a normal entity text searched or selected (Figure 2).

To focus on particular entities, HyperStorylines allows users to **filter** and/or **select** them. An entity is filtered if the user double clicks on it or searches by its name in the search bar. This will change the view to only show the chosen entity and all its connections. The filtered entity will be surrounded by a blue rectangle (*e.g.,* Figure 3). To select an entity, the user only needs to click once on an entity. This will make its name and line blue, render all related entities dark gray and fade-out all those that have no relationship with it. Scrollbars (Figure 2) will indicate where to find filtered and selected entities, in addition to expanded relationship bars.

Software Implementation

HyperStorylines is developed in Javascript using D3.js[42]. We use the D3 narrative layout[43] for computing storyline geometry, which uses jLouvain[44] to detect communities (clusters) and position groups of characters/storylines in the layout. Django[45] is used as the Web server framework. Text/keyword search in the original articles is implemented using ElasticSearch[46].

Feedback from journalists and support staff

The second workshop gave journalists (J1,J2) and support staff (S3,S4) the opportunity to view early versions of our designs. One last 1h follow-up session with one support staff validated our final design. Organizing additional work sessions with journalists became difficult in 2020 as they were over-solicited covering the Covid-19 crisis and critical worldwide socio-economical events.

Their feedback refined HyperStorylines and influenced our design decisions in the following ways:

- Both journalists and staff appreciated the focus on relationships, and commented on how it was *"very efficient to identify potential relationships at a glance"* (J1), but that it is also important for them to investigate the origin of the relationship (J1,J2). That is why when a viewer clicks on a relationship they can see the original text of the articles they originated from, with the relevant entities highlighted.

- In our initial prototype all types of entities were presented in a single view (as do PAOHvis[7] and TimeArcs[24]). Nevertheless, our journalists (J1,J2) commented that seeing all types of entities together can be overwhelming. Instead they commented that their questions focus first on identifying relationships between people, and other entities are seen as explanations or details of the relationships. They thus suggested starting with views that focus on people only. This motivated our "nested" approach, which allows users to open up relationships to see more information. An exception to this are entities that are search terms. These should always be visible irrespective of the view configuration.

- One journalist commented that it would be nice to know if some relationships are more likely than others, *e.g.,* if a relationship seems to come from multiple sources/articles (J2). As our data come from an automatic extraction process, the number on top of a relationship is by default the number of nested entities. But the semantics of the number can be mapped to other information associated with the relationship, such as the number of articles that have generated the relationship (the more articles, the more likely the relationship).

- We used a small set of colors in our visualization (variations of gray and black for the main visualization, aqua for selections and rust for keyword entities). The reason behind it is that journalists (J1,J2) gave examples where they would like to use color as a means to characterize entities further (*e.g.,* by gender, political party, *etc.*).

- While all four participants acknowledged that the focus of the tasks discussed is on specific political entities, that can be small in number, they mentioned that having an overview of all the entities in the dataset would provide some context. This led to the addition of histograms in the final tool.

Comparative Study

Besides the feedback from journalists and journalistic staff on different iterations of the HyperStorylines prototype, we also evaluated the final tool under the tasks identified in these workshops. We were particularly interested in how difficult it is to do these tasks with little training, and what types of strategies users adopt to solve the tasks. We focused on the participants' understanding of the visual representation of the hypergraph as the HyperStorylines visualization is simple in its concept, but the prototype incorporates several possible views. We also took this approach because the features of the overall system in which HyperStorylines is embedded might vary depending on the application domain. For example, hypergraphs with an important geographical component may require a map view of the location entities. In this section we describe: the baseline visualization we selected to compare HyperStorylines with; the modifications we did to each tool to address the points above and to make them comparable; the tasks we used to make this comparison; the hypotheses we formulated before conducting it; as well as the study design, procedure and details.

Tools: HyperStorylines and adapted PAOHvis

Here we describe the baseline tool we compared HyperStorylines against, as well as the adjustments in functionality we made in order to make the two tools comparable.

PAOHvis As explained in Section Related Work, from the available visualizations for dynamic hypergraphs at the time we designed our study, PAOHvis[7] was the most competitive and appropriate one for us to use as baseline. This is because it can accommodate multiple types of entities and their connections (evolving over time), it is one of the most recent suggested visualization focusing on similar data, and it included an evaluation. Moreover, it is simple to understand and it reveals all possible relationships at a glance (something that HyperStorylines does not do).

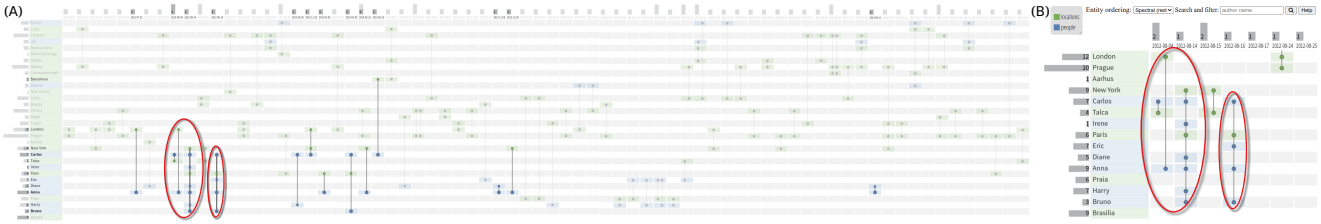

Figure 5. PAOHvis using the dataset in Figure 1. The red circles indicate the same relationships (hyperedges) highlighted in that figure. (A) Shows an overview of the dataset, and (B) shows a zoomed view of the circled relationships in addition to the available UI selectors used in the user study. Notice how this view is more cluttered than Figure 1, despite showing the same number of entities.

We initially also considered TimeArcs[24] as a potential baseline, that also shows entities colored by type and as straight rows/lines (similar to PAOHvis), but connections are shown on hover and in the form of arcs. An early pilot study revealed that following relationships was harder when shown as arcs (compared to the straight lines in PAOHvis) and the TimeArcs interface was too cluttered when many relationships were visible, as each relationship between more than two entities exponentially increased the number of arcs (to connect each entity with all the others). We thus chose PAOHvis as the state of the art technique to compare against.

In terms of visual mapping, `PAOHvis` is roughly organized in a matrix layout (Figure 5-A). Each entity is represented by a row colored by the group it belongs to (which is user defined). In our study, we use color to represent the type of entity. The first column shows the name of each entity, while the other columns represent dates. Each hyperedge is represented by a vertical black line that connects with a dot the entities involved in a relationship, to differentiate them from rows of entities crossed by the line that do not take part in the relationship. At the left of the entity's name lies a bar whose length and inner number show a particular statistical indicator (for example, the number of hyperedges the entity participates in). The dates at the top have a similar statistics bar.

`PAOHvis` displays all entities and hyperedges/relationships in its initial view and can consume much screen real-estate (Figure 5-A). It thus allows users to select subsets of entities to highlight / filter (either by text search or by clicking on them).

Modifications to PAOHvis. In order to reduce confounding factors and to focus on the visual mapping of the hypergraphs, we made some modifications from the original `PAOHvis` code[47] to ensure it had similar functionality to `HyperStorylines`. First of all, we replaced the original navigation menu with a simpler one that only allowed participants to reorder the entities on the vertical axis, and to search and filter (Figure 5-B).

Second, the original text search in `PAOHvis` retrieved entities whose names matched the first part of any of the input words. We replaced this search with one that retrieves entities by a full match of the input text. This behavior is consistent with `HyperStorylines` and makes tasks performed in the study easier by reducing the search results returned. Additionally, when no results were found, the original interface would show the entire dataset. This can be confusing for our tasks. We thus show an empty visualization instead (consistently with how `HyperStorylines` behaves).

`PAOHvis` allows users to select the type of statistics to show for entities and dates. We removed these selectors and set that indicator to the number of relationships associated with the entity or the date, as this is useful for many of our tasks (*e.g.,* finding the number of relationships an entity has with another). Finally, we fixed some options by default like: color entities by their type, and order edges by length inside the time columns to cluster entities involved in a hyperedge together as much as possible.

All user input (interaction) was the same across the two tools (*e.g.,* text content was copied in the clipboard when highlighted, double click filtered, scrollbars allowed to pan, scroll-wheel allowed to zoom in/out). The only difference was that selecting several entities in `PAOHvis` requires users to press ctrl or shift while clicking on them, which was part of the original `PAOHvis` design.

HyperStorylines modifications To further ensure the two visualizations are as equivalent as possible, we removed from the original HyperStorylines tool the histograms of entity and time frequencies, the view of the original text of news articles (that relationships come from), the keyword search for terms inside the original articles that creates special storyline entities, and the boxes that show the search history.

In particular, we removed the option to aggregate as it influences at which level the data is displayed and manipulated. For example, aggregation in `HyperStorylines` allows users to group hyperedges by month or year for easier analysis of their evolution over time, which would make the comparison with `PAOHvis` unfair. Moreover, such an aggregation goes beyond the visual representation of a hypergraph *per se*. To ensure a fair comparison we would have had to extend `PAOHvis` to support time aggregation, which would have entailed an extensive redesign of that tool.

Tasks

The tasks supported by `HyperStorylines` are related to both dynamic hypergraphs and set visualizations. However, as we are motivated by high-level questions on relationships derived from investigative journalism, tasks related to the evolution of graph topology[21,48] or set operations[49] were too low level or did not match our context. We thus took a similar approach to Kerracher *et al.*[50] and structured the tasks identified by our users using Andrienko & Andrienko's task taxonomy[51].

First of all, in this taxonomy tasks are categorized in two reading levels: *elementary* when they refer to individual elements of the dataset, and *synoptic* when they involve the whole dataset or a subset of it. We ensured that our tasks

would require participants to do both types of reading with the tools. For each of these reading levels, the taxonomy has three types of tasks: *look-up*, *compare* and *relation-seeking*. Taking into account these reading levels and task types, we structured the questions derived from those of journalists (Section Workshops) in the following concrete experimental tasks that we used in our evaluation:

T1: `Find relationships` (**Q1** in Section Workshops): This task requires users to search for relationships between several entities and is a key need identified in our workshops with journalists. This is an *elementary look-up task* in Andrienkos' taxonomy[51], in which one or more references (entities) are given to find a particular goal/answer. This task is similar to T1 in the PAOHvis paper[7].

T2: `Characterize relationships` (**Q3** in Workshops): In this task we ask participants to characterize how the number of entities connected to a particular entity has evolved over time (increased, decreased, not changed, unclear). It is motivated by the types of patterns journalists look for in order to compare entities, although in our case we focus on identifying the pattern (instead of asking for a comparison among patterns). We made this choice so that our task is similar to T5 in the PAOHvis paper[7]. This is a *relation-seeking task*[51], where participants have to first look-up an entity and then characterize a pattern over time.

T3: `Find similar entities` (**Q3** in Workshops): This task can encompass several variations, as the definition of similarity may depend on the context. We focused on relationship patterns, asking participants to find if different entities had similar types of connections (in our case if their most connected person is the same). This is a *relation-seeking task*[51], that is composed of two subtasks. First, for each of the references (entities) given in the question, it is necessary to *look-up* the entity they share the larger number of relationships with, and then *compare*. This task is similar to T2 in the PAOHvis paper[7].

T4: `Find large relationships` (**Q4** in Workshops): This task requires users to identify events that involve a large number of entities. This task is a *synoptic comparison task*[51] that requires getting an overview of the dataset, comparing the number of entities involved in each relationship and then selecting the one with the maximum number of entities. As far as we know this task has not been considered before.

Task Variations: Additionally, in the Andrienkos' taxonomy[51], tasks are composed of a *target* (what we are looking for) and *constraints* (known information to find the target). We used this notion of constraints to vary the complexity of our tasks. The entities that characterize a relationship are often of different types (**Q2** and **Q3** in Section Workshops). We thus varied the constraints of the tasks such that participants had to gather information using entities of one type or of multiple types. To make the study duration tractable, we limited the number of types of entities to only three (even though both tested visualization techniques support more). These types are related to spatio-temporal aspects of the relationship: people, locations and time. The combination of these three entity types led us to three variations for our tasks, increasing in complexity as more entity types are involved:

- **Single type of entity in time:** these are questions about one or more people over time. In `HyperStorylines`, this requires participants to maintain the default view and follow the lines of one or more entities. In `PAOHvis`, this level requires participants to focus on entities of the same color.

- **Two types of entities but no time:** these are questions about relationships between people and locations. In `HyperStorylines`, this variation requires participants to see locations explicitly, either by changing the default view or expanding relationships. In `PAOHvis` it requires participants to focus on entities with two different colors.

- **Two types of entities in time:** We expected this level to be the most difficult, as it requires participants to search for people, locations and time. In `HyperStorylines` this level may require changing the default view, and participants have to focus on entities on both axes (story entities and related-by entities) and also expand relationships (to see the nested ones). In `PAOHvis` this level requires focusing on entities with two different colors, as well as to read the dates at the top of the visualization.

For each task, we created three questions, one for each task variation of entity types and time, summarized in Table 2.

Hypotheses

We formulated the following hypotheses for each of the task categories:

- **H1:** For tasks `Find relationships` (T1) and `Characterize relationships` (T2), we expect `HyperStorylines` to perform better. These tasks require participants to focus on either the entire history (or a subpart) of two or more entities. In `HyperStorylines` the histories (curved lines) of entities are constrained between the first and last date of any relationship they participate in, which we expect will make the comparison easier. On the other hand, in `PAOHvis` the rows of each entity span the entire time scale, making it likely more time consuming to follow the rows for each entity, and likely to lead to more errors and lack of confidence.

- **H2:** For task `Find similar entities` (T3), we expect `PAOHvis` to perform better as it shows all the entities of all types at once, while in `HyperStorylines` participants may need to change views to find the right answer.

- **H3:** For task `Find large relationships` (T4), we expect participants to perform better with `HyperStorylines` than with `PAOHvis`. We expect this because in `PAOHvis` participants will likely find it harder to find long vertical lines, see how many entities are attached to them, and then compare them across different dates. On the other hand, the lines of entities in `HyperStorylines` explicitly converge together and increase the height of relationship bars. Thus, longer relationship bars will be easier to identify (and compare).

- **H4:** We expect that with `HyperStorylines` participants may take some time to understand how to construct the right view to answer questions, especially for questions that consider a third entity (locations). On the other hand `PAOHvis` always has the same view. We thus expect some learning to occur, which we test in two phases (exploratory and advanced). Participants will take longer to complete tasks with `HyperStorylines` while they are still learning the tool (in the exploratory phase) than with `PAOHvis`. However, we expect that as they gain more

	People + Time	People + locations	People + locations + time
Find relationships **H1a:** `HyperStorylines` performs better	In which dates were person X and Y mentioned together?	In which unique locations were person X and Y mentioned together?	In which date was person X mentioned in location Y?
Characterize relationships **H1b:** `HyperStorylines` performs better	Over time, how did the number of people involved in the relationships of person X evolve?	Over time, how did the number of people involved in the relationships in location X evolve?	Between dates A and B, how did the number of locations involved in the relationships of person X evolve?
Find similar entities **H2:** `PAOHvis` performs better	Is the person that appears the most with person X and same that appears the most with person Y?	Is the person that appears the most in location X and same that appears the most with location Y?	Between dates A and B, is the location that appears the most with person X and same that appears the most with person Y?
Find large relationships **H3:** `HyperStorylines` performs better	On which date was the event with the maximum number of people?	On which date does a relationship between locations X and Y involve the maximum number of people?	Between dates A and B, on which date are the locations X and Y involved in relationship with the maximum number of people?

Table 2. Examples of questions asked, by task (row), and by variation of entity types and time (column). Associated hypotheses are under each task.

experience (in the advanced phase) they should feel more comfortable and finish them faster. We explain these two phases in more detail in the next section.

Although we expect there are differences across the two techniques, we note that these are complex analytic tasks, and the two visualizations require different interaction strategies to answer questions (*e.g.,* `HyperStorylines` may require view changes, whereas `PAOHvis` requires scrolling). It is thus unclear if there is a speed/accuracy trade-off. Nevertheless, we do expect differences on one of these performance aspects, as well as in participant confidence.

Experimental Design and Procedure

We use a within-subjects design where all participants are exposed to each visualization in turn (counterbalanced across participants using a Latin square). Given the current Covid-19 pandemic we conducted the study remotely by using the Jitsi video conference application[52] and a Web-based interface. It lasted approximately two hours and a researcher was (virtually) present the whole time. Participants were encouraged to think aloud and the sessions were video recorded. Participants signed a consent form and sent a digital copy by email. They then filled in a demographics questionnaire, and were redirected to a web page with an overview of the experiment. After this, the first visualization to be used was explained and participants were able to interact with it until they had no further question. We did not train participants on specific tasks in this first view of the tool. The main study consisted of two phases, the Exploratory phase first, followed by the Advanced phase. The procedure was repeated for the second visualization.

1) Exploratory Phase. The goal of this phase is to study how participants intuitively formed strategies for solving the trials and how effective these were, without providing them with any particular training (other than the basic visualization functionality). This phase consisted of 12 trials (3 variations for each task). After submitting an answer, participants received feedback on whether it was correct or not. In the case of wrong answers, the tool indicated which was the correct one. In addition, the experimenter would ask them to repeat the trial, where they explained possible strategies to solve the task. In the results section we only report measures from the first time they tried each trial (*i.e.*, before system feedback). After each trial, participants reported their confidence and how easy they felt it was to complete it. The order of trials was such that it increased in terms of difficulty, starting with the least complex variations (Section Tasks). Once all trials in this phase were answered correctly, participants were able to continue to the next one.

2) Advanced Phase. The goal of this phase is to capture *more experienced* user behavior of the participants, as we assume the exploration phase helped them develop strategies and understand the subtleties of each visualization. This phase contains another 12 trials, except that participants did not receive feedback from the experimenter or the web application about their strategies or the correctness of their answers. Moreover, trial order was randomized.

Once the two phases were completed for both visualizations, participants were redirected to a post-hoc questionnaire in which they rated, on a Likert scale, how hard it was to answer questions in each of the 4 task categories with each visualization and to justify their answer. Finally, they were asked to select which visualization they preferred overall.

In total, the experiment consists of 6 participants (discussed later on) $\times$ 2 visualizations $\times$ 2 phases $\times$ 4 tasks $\times$ 3 task variations as repetitions = 288 trials.

Datasets

As the tasks we evaluated were high level, we were interested in studying the usability of the tools with a realistic setup. We thus decided to use real datasets to create the questions for the Advanced phase. We created 12 datasets, one for each question, that are subsets of the original dataset extracted from news articles provided by Ouest France (see Section HyperStorylines). The original dataset is composed of 8653 articles dated from January 3rd, 2012 to the December 31st, 2018. It has a total of 33858 entities: 24554 people, 2361 locations, 5040 organizations and 1903 dates (time). We anonymized all entities, either by using the Faker Python library[53] in the case of people and locations, or by applying a random shift in the dates.

Each of the generated datasets contained 1 year of news articles, either starting from the beginning of the year (*e.g.,* from January 1st to December 31st, 2012) or from the middle (*e.g.,* from July 1st, 2012 to June 30th, 2013). In order to

reduce their complexity, we first filtered out relationships that had more than 100 entities of the same type and kept the 500 most connected entities of type people and locations, *i.e.*, those that participate in the larger number of relationships. We then applied a second filter to remove all relationships that included less than 3 entities or had no date, as these would not show as a relationship in `PAOHvis`. Finally, to remove big distractors for the task `Find large relationships`, we ensured that there would be only one event with the maximum number of people, and all the rest had at most half the number of people of that maximum. We included this constraint of making the large relationships significantly larger than other relationships in order to focus the task on the search of the more general pattern, and to ensure participants would not need to spend time counting the exact number of entities between several possible candidates in order to compare them. The final datasets have, on average, a mean of 529 entities (SD=27) and of 182 relationships (SD=74).

To ensure that participants would not remember answers between visualizations, each question had two possible datasets: the original one and a mirrored version. Which dataset was used with PAOHVis or with HyperStorylines was determined using a Latin Square. In this mirrored variation we first reversed the order of the dates of the original dataset, so the first relationships will appear last and the last will be at the beginning. We also anonymized all the entity names and dates for the timeline. This ensured that the structure, patterns, and frequency of relationships for each question were consistent across visualizations, but the entity names in the questions (and their answers) would look different.

For the Exploratory phase, we used a single (smaller and less complex) dataset and its mirrored version, that were alternated between one question and the next. If a trial was performed using the original training dataset, the next one was performed using the mirrored version. This dataset was created using the co-authorship dataset from PAOHvis[7] available on its website.

Apparatus and Participants

As our participants performed the study remotely, we could not ensure they had the same screen and laptop. We stored the participants' screen resolution, but did not see any effect of resolution on performance (we note however that all participants had different resolutions and did not control explicitly for their screen size - plots provided in supplemental material). The study's UI was developed as an extension of the Django-based back-end of the HyperStorylines tool, where we also embedded PAOHvis. We ensured that inside the study interface each visualization filled all available screen real-estate, leaving only space for question & answer input at the top. Additional screenshots of the user interface for both visualizations are provided in the supplemental material.

As mentioned earlier, access to journalists became quite difficult in 2020. Given that `HyperStorylines` is a general purpose visual analysis tool, we decided to evaluate it with participants who are knowledgeable in visualization design, so as to not only observe objective measures such as time and errors, but also collect feedback on usability and critiques about our design choices. Participants knowledgeable in visualization and interface design are accustomed to using and critiquing complex interfaces and can thus be considered as "best case" for understanding designs: if aspects of the interfaces are hard to use for them they will also be challenging for the general population, and the best performance they attain is likely indicative of what expert users will be able to perform.

We recruited 6 participants (4 male and 2 female), who were all HCI or visualization experts with between 3 and 16 years of practice. We consider this number of participants sufficient, as HCI and visualization studies often have small numbers of participants but with relevant results[54,55] and there is no magic number of participants[56]. When it comes to statistical evidence, our method of using CIs can still provide evidence of differences with even two participants[57]. All participants had normal or corrected-to-normal vision and their age ranged from 26 to 45 (M=32, SD=7). They were all volunteers, and did not receive any monetary compensation.

Measures

We collected four primary measures (2 objective, 2 subjective) for all the tasks and the two phases:

- `Completion Time`: measured from the moment participants see questions until they validate their answer.
- `Error Rate`: percentage of incorrect answers per task.
- `Self-reported Confidence`: on a 5-point Likert scale, from *highly confident* (5), *confident*, *neutral*, *not confident*, to *not confident at all/random selection*(1).
- `Self-reported Easiness to Complete Task`: on a 5-point Likert scale from *very easy* (5) to *very difficult* (1).
- `Overall Preference`: we also asked participants which visualization they preferred the most at the end of the experiment. This overall preference question was accompanied by a text field for participants to provide additional explanations, feedback on usability and strategies.

Results of Comparative Study

We report and interpret all our results using interval estimation instead of p-values[57,58]. We report sample means of 95% confidence intervals (CIs), which means we are 95% confident that this interval includes the population mean. We construct all CIs using BCa bootstrapping (10,000 bootstrap iterations). We analyze the CIs using estimation techniques, *i.e.*, we interpret them as providing different strengths of evidence about the population mean, as recommended in the literature[57,59–62]. When reading a CI of *mean differences*, if the CI does not overlap with 0, this is evidence of a difference, corresponding to statistically significant results in traditional p-value tests. Nevertheless, CIs allow for more subtle interpretations. The farther from 0 and the tighter the CI is, the stronger the evidence. Equivalent *p*-values can be obtained from CI results following Krzywinski and Altman[63]. Additionally, we present a summary of the strategies used by participants, as well as their comments while conducting the tasks. This information was gathered using a two-step coding process by one of the authors. As a first step, we collected the possible strategies for each task and each visualization as open coding from the initial pilots of the study. These codes were then used as input to the second step to guide the analysis of the strategies used by

Figure 6. Overall results for the Exploratory phase (top) and Advanced phase (bottom). The first column reports mean Completion Time in seconds for all tasks per visualization (left) and mean differences between visualizations (right). The second column reports mean Error Rate in % for all tasks per visualization (left) and mean differences between them (right). Error bars for the first and second columns represent 95% Bootstrap confidence intervals. Evidence of differences are marked with a ⊛ (the further away from 0 and the tighter the CI, the stronger the evidence). The third column shows the percentage of trials that participants reported being from `highly confident` (■) about, to `highly not confident/random selection` (■). The fourth column shows the percentage of trials that participants reported being from `Very easy` (■) to complete, to `very hard` (■).

participants in the main study. The coder would determine if the way participants performed a task followed a previously-coded strategy or not. When the strategy was different from the pre-coded ones, the coder would note in more detail the new strategy. Additionally, the coder would write down the comments participants said out loud as well as general observations about their behavior when relevant.

Collected data, analysis scripts and dataset generation code are available as supplemental material to this submission [†]. In the text we report means and trends. The supplemental material also include detailed CI values. All analyses were planned before data collection.

Performance and Learning across phases

Completion time: The first column of Figure 6 shows completion time for all tasks collectively for both phases. When looking at all tasks together, there is no evidence of difference between tools in the exploratory phase (mean is 92.44s CI [73.03, 124.45] for `HyperStorylines` and 93.39s CI [82.91, 120.85] for `PAOHvis`). However, there is evidence that in the advanced phase `HyperStorylines` was faster by 17.51s CI [4.46, 35.59], with a mean time of 67.16s CI [55.82, 78.87] for `HyperStorylines` and 84.67s CI [70.57, 99.72] for `PAOHvis`. Indeed, there is evidence of a learning effect (**H4**) in `HyperStorylines`, with participants being on average by 25.29s faster with in the advanced phase (CI [12.78, 46.87]), but no evidence for `PAOHvis` 8.72s (CI [-0.98, 21.13]). Please refer to the supplemental material for additional plots that show this learning effect.

Error Rate: There is no evidence of difference in error rate between the tools for either phase (second column of Figure 6). Nor any evidence of a decrease in error rate between the exploratory and advanced phases.

Self-reported confidence: Overall, confidence was high for both visualizations in both phases (third column of Figure 6). Although this confidence was higher in `HyperStorylines` than in `PAOHvis`. For `HyperStorylines` participants reported a high confidence in 86% of the tasks ($M = 4.24$, $SD = 0.83$) in the exploratory phase and in the advanced ($M = 4.19$, $SD = 0.78$). For `PAOHvis` participants reported a high confidence in 81% in the exploratory phase ($M = 4.13$, $SD = 0.98$) and 71% in the advanced ($M = 3.83$, $SD = 1.01$).

Self-reported easiness: Similar to self-reported confidence, the perceived easiness was high for both visualizations in both phases (fourth column of Figure 6). In the exploratory phase both tools reported similar values: participants reported 66% of the tasks a high easiness to do a task ($M = 3.94$, $SD = 1.0$), and 69% for `PAOHvis` ($M = 3.81$, $SD = 1.0$). However, in the advanced phase easiness is higher with `HyperStorylines` 79% of the tasks were perceived with high easiness ($M = 4.03$, $SD = 0.79$) than with `PAOHvis` (64% of the tasks were perceived with high easiness, $M = 3.69$, $SD = 0.94$).

Overall Preference: These results are consistent with overall preference and feedback provided by participants after having experienced both tools. From the six visualization experts, five preferred `HyperStorylines` overall. In their comments, most participants agreed that although sometimes it was hard to select the appropriate view with `HyperStorylines` or that it required more interactions to create it, it was preferred for most of the tasks. P3 declared in the post questionnaire: *"I liked HyperStorylines because I could change the axis to fit my needs of counting stuff. This gave the impression that I was simplifying a complex visualization and then looking at an easy to understand visualization. I have more manipulating power over the tool"*. P4 and P6 had similar comments. P5 said that for them it was a trade-off between trying to plan the views in `HyperStorylines` and ending with a simplified view appropriate to their task. They added that even though `PAOHvis` required less actions to complete the tasks, they had to scroll a lot which was cognitively harder. P1, who preferred `PAOHvis`, stated that one of the problems with `HyperStorylines` was that sometimes they felt lost as to what view they were in.

Performance per task

Next, we break down our results across the four tasks. For this analysis we focus on the Advanced phase where

[†] https://ilda.saclay.inria.fr/hyperstorylines

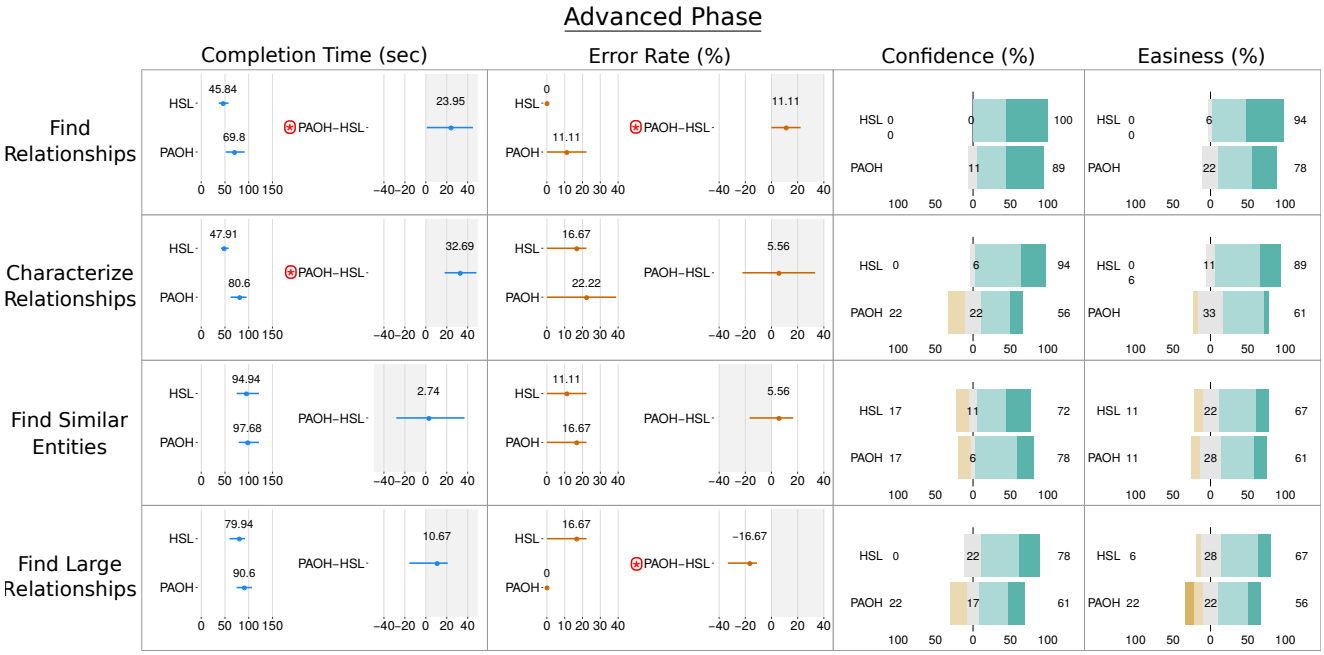

Figure 7. Results for Completion Time (sec), Error Rate (in %), self-reported confidence and self-reported easiness to complete the task for each task in the Advanced Phase. In each row (task) for the first two measures, mean values per visualization are seen on the left and means of pairwise differences on the right. Error bars represent 95% Bootstrap confidence intervals. Gray rectangles indicate the direction of our hypotheses. Evidence of differences are marked with a ⊛ (the further away from 0 and the tighter the CI, the stronger the evidence). The third columns shows the percentage of trials that participants reported being from `highly confident` (■) about, to `highly not confident/random selection` (■). The fourth columns shows the percentage of trials that participants reported being from `Very easy` (■) to complete, to `very hard` (■).

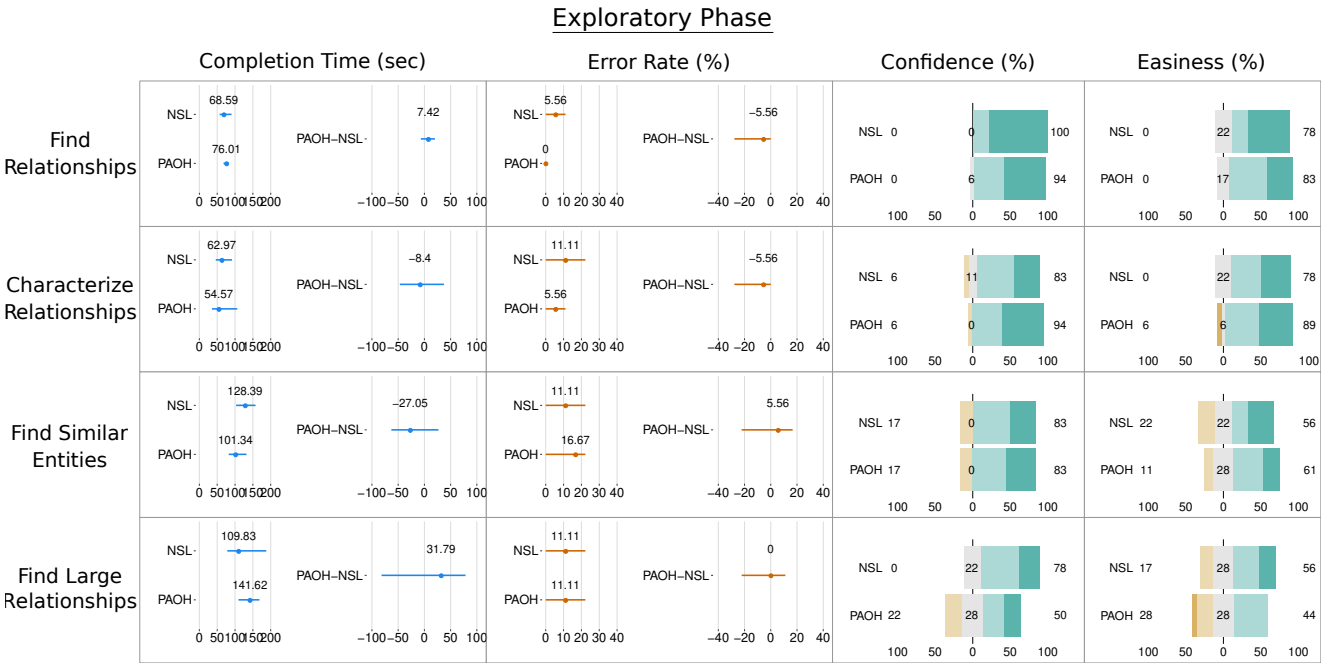

Figure 8. For completeness, we present here results for Completion Time (sec), Error Rate (in %), self-reported confidence and self-reported easiness to complete the task for each task in the Exploratory Phase. In each row (task) for the first two measures, mean values per visualization are seen on the left and means of pairwise differences on the right. Error bars represent 95% Bootstrap confidence intervals. The third column shows participants confidence, as percentages of trials that participants reported being from: `highly confident` (■) about to: `highly not confident/random selection` (■). The fourth column shows the percentage of trials that participants reported being from: `Very easy` (■) to: `very hard` (■) to complete. We do not stress differences in this phase as the experimenter could intervene to aid participants.

participants can be considered as more experienced users, but we report any learning effects between the two phases if there is evidence of it. Figure 7 shows a summary of our results related to our main hypotheses in the Advanced phase. Plots for learning effects per task are in the supplemental material (indicated with *cf. Sup* in the text). For the sake of completeness Figure 8 also shows the results per task in the exploratory phase. However, we do not stress these results in the text, as the experimenter intervened while participants completed these tasks.

Our main hypotheses do not consider internal task variations of entity type and time. Nevertheless if we observe any trends that change depending on task variation we briefly report them. Plots for this additional analysis on differences between task variations are in the supplemental material.

`Find relationships` *(H1a: HSL>PAOH)* There is weak evidence of differences in both Completion Time and Error rate between the tools. For `HyperStorylines` the completion times were lower than in `PAOHvis` by 23.95 CI [0.91, 45.02], with `HyperStorylines` having a mean time of 45.84s CI [36.81, 57.46] and `PAOHvis` having 69.8s CI [51.73, 91.39]. Similarly, `HyperStorylines` was less error prone than `PAOHvis` by 11.11% CI [0, 22.22]. The error rate for `HyperStorylines` was 0% CI [0, 0] and for `PAOHvis` was 11.11% CI [0, 22.22].

The trends for error and time are consistent across variations, but we observed that `HyperStorylines` was mainly faster in the complex task variation that combined 2 types of entities (people+location) over time (cf. Sup).

Looking at potential learning effects for this task (cf. Sup), there is evidence of learning for `HyperStorylines` as participants were faster in the Advanced phase than in the Exploratory one by 22.85s CI [15.30, 33.46], but no difference for `PAOHvis` (faster by 6.22s CI [-15.35, 26.56]). We also have weak evidence that `HyperStorylines` may be less error prone in the Advanced phase by 2.78% (CI [0, 13.89]), while `PAOHvis` was more error prone by 11.11% (CI [0, 22.22]).

Both self-reported confidence and easiness to complete the tasks were higher for `HyperStorylines`.

In terms of strategies, we observed that all participants used the search/filter and selections features to perform this task. We also observed that with `HyperStorylines` most participants (5/6) were able to construct an appropriate view or expand a relationship when the questions required them to search for a third entity. P1 was the exception: for them, the construction of the views was confusing, as they stated in the post-questionnaire: *"[...] Using [Hyper]Storylines I often forgot what was represented where, or did not know how to optimally choose the axis"*. Half of the participants (P4, P5 and P6) also commented that they had to scroll more in `PAOHvis` to find particular relationships. For example, P4 mentioned in the post-questionnaire: *"On both [tools] the task was easy except that for PAOHVis, because dates are not filtered out (as opposed to [Hyper]Storylines) when using filters, tasks that involve finding dates require manually scrolling through the complete timeline."*.

T1 Summary: Results in subjective and objective performance measures support **H1**, with `HyperStorylines` performing better across all measures and task variations about finding relationships. They also support **H4** on the higher amount of learning for `HyperStorylines`. Participants commented that the condensed `HyperStorylines` view aided them. With a single exception, they had no problem constructing the appropriate view. We do not observe any trade-off between speed and accuracy for this task. On the contrary, it seems that the condensed view of `HyperStorylines` allowed participants to be both faster and make less errors than with `PAOHvis`.

`Characterize relationships` *(H1b: HSL>PAOH)* There is strong evidence of a difference in Completion Time with `HyperStorylines` being faster by 32.69s CI [17.91, 48.32] than `PAOHvis`. `HyperStorylines` had a mean completion time of 47.91s CI [41.73, 57.89] and `PAOHvis` had 80.6s CI [61.91, 95.48]. There is no evidence of differences in Error Rate.

The trend for `HyperStorylines` to be faster is consistent across task variations (cf. Sup). When it comes to error, although there is no difference overall, we observed that `HyperStorylines` was more correct in the complex variation that combined 2 types of entities (people+location).

There is evidence of a positive learning (cf. Sup) in terms of time only for `HyperStorylines`, which becomes faster in the Advanced phase without any difference in accuracy. We do not observe any learning effect for `PAOHvis` between the two phases, on the contrary, we had weak evidence of it becoming more error prone in the Advanced phase without a trade-off in time.

Both self-reported confidence and easiness to complete the task were higher for `HyperStorylines`.

Similar to the previous task, half of the participants (P1, P4 and P6) mentioned that they felt they had to scroll less with `HyperStorylines`, as filtering by a particular entity reduces the timeline to show only the dates associated with that entity. In particular, P6 commented that with `PAOHvis` *"given that all the dates are visible, it gets confusing to try to get a pattern using the whole time"*. We also observed that although participants used the number of nested entities on top of each relationship, 2 out of 6 participants (P1 and P2) would not completely trust or understand them and opened to confirm their meaning. For example, in the question that involved both people and locations, P1 opened the relationship to confirm that the number of nested entities was the same as shown at the top.

T2 Summary: Thus, we have support for **H1** that expected better performance with `HyperStorylines` when it comes to the Characterizing relationships tasks, with respect to completion time and subjective measures across task variations. We do not observe a trade-off between speed and accuracy, as participants took less time with `HyperStorylines` than with `PAOHvis` without sacrificing accuracy. Participants' comments highlight that the more condensed `HyperStorylines` view made the task easier, which could explain this difference in performance. Results also further support **H4** on the higher amount of learning for `HyperStorylines`.

`Find similar entities` *(H2: PAOH>HSL)* There is no evidence of differences between the two tools, neither for Completion Time nor for Error Rate.

When looking at task variations, there is no difference when it comes to time (cf. Sup). Nevertheless, when it comes

to errors we see different trends. HyperStorylines is more correct in variations that consider only one entity type over time, and only two entity types (people+location) without time. For the more complex variation of two entity types over time PAOHvis is more correct, so the expected difference in our hypothesis is only visible here.

There is also strong evidence of a learning effect in terms of time for HyperStorylines only (cf. Sup).

Subjective metrics are similar between tools and high for more than 50% of trials for both tools.

We did not find special differences for the strategies followed using the two tools. For PAOHvis we observed that P1, P3, P4 and P6 used the statistics view of PAOHvis to count the number of relationships of each person to compare them later. In the post-questionnaire, P1 and P3 had positive comments about how this feature helped them for this task, while P4 commented that its usefulness depended on the task: "*[...] When no time range is involved, the grey bars with counts of PAOHVis make the task easier than in Storylines but when dates are involved, the length of the unfiltered timeline in PAOHVis makes it more difficult than in Storylines*".

T3 Summary: There is thus no support for **H2** that expected overall better performance for PAOHvis, but performance seems to differ across task variations. Results further support **H4** on the higher amount of learning for HyperStorylines. Comments from participants seem to be divided between the two techniques.

Find large relationships *(H3: HSL>PAOH)* There is strong evidence that PAOHvis is less prone to errors than HyperStorylines for this task by 16.67% CI [11.11, 33.33]. PAOHvis has a mean error of 0% CI [0, 0] and HyperStorylines has 16.67% CI [0, 22.22]. There is no evidence of difference in Completion Time between the tools, although HyperStorylines is faster by 10.67s CI [-15.74, 20.91]. Completion time for HyperStorylines is 79.94s CI [59.61, 92.05] for this task and 90.61s CI [74.46, 106.75] for PAOHvis.

When looking at the tasks variations individually (cf. Sup) we see that HyperStorylines is faster when considering two entity types (people+location) without time so the expected difference in our hypothesis is only visible here. Regarding errors, PAOHvis is more correct only in the simplest variation of one entity type over time.

There is strong evidence of a learning (cf. Sup) in terms of time between the two phases for both PAOHvis and HyperStorylines. Also, there is evidence of learning in terms of error rate for PAOHvis.

Subjective metrics are high for more than 50% of the trials for both tools. However, for HyperStorylines both self-reported confidence ($M = 4.06$, $SD = 0.72$) and easiness ($M = 3.78$, $SD = 0.81$) are higher for HyperStorylines than with PAOHvis ($M = 3.61$, $SD = 1.09$ for confidence and CI $M = 3.39$, $SD = 1.24$ for easiness), showing that for this task objective and subjective metrics were not aligned.

When using PAOHvis, all participants expressed that it was hard to follow the lines to understand which entities were involved in which relationships. All of them tried to reduce the length of the relationships by reordering the entities on the left. P2 commented in the post-questionnaire that "*[...] with PAOH it was harder, especially to count the number of people, as the people and entities are mixed on the same axis and people at the same event are not all the time close from each other*". P1, P3, P4 and P5 made similar comments. Regardless, half of the participants made mistakes in this task when using HyperStorylines which can be attributed to the construction of an inefficient view. In the case of P1 and P3, the suboptimal views aimed to find the large relationships by searching for the largest number on top of each relationship instead of the longest relationship, which led them to answer incorrectly. P6 was also confused while constructing an appropriate view and commented while conducting the task: "*as I can't see the complete visualization I always feel I'm missing something*". This confusion also led them to answer incorrectly.

T4 Summary: Our results do no support **H3**, which expected better performance for HyperStorylines. On the contrary PAOHvis is less error prone, but this is the case only for the most simple task variation. We do not observe a trade-off between speed and accuracy as there is no difference in completion time. A learning effect **H4** was observed with both tools. Participants' comments highlight that for this task constructing the appropriate view in HyperStorylines was challenging and this could explain their reduced accuracy.

Summary of Results and Discussion

We now summarize and discuss our findings, both the early feedback from journalists (see Feedback), and the results of our comparative study between HyperStorylines and PAOHvis (see Results).

Journalists' feedback from an early version of the entire HyperStorylines system was very positive, helped us refine our design choices and confirmed that analysts liked the ability to progressively construct partial views and see nesting relationships. Their feedback stressed the potential value of the visualization for helping them answer their questions. Moreover, it helped us precisely identify the questions they seek to answer when exploring relationships between entities, that became the basis of the tasks used in our comparative study.

From the tasks we tested in that study, we confirmed that becoming an experienced user of HyperStorylines as a tool takes some effort (**H4**). In a first exploratory phase, where participants had received neither instruction nor practice for completing the tasks, we observed in HyperStorylines a learning effect across all tasks, that was present in PAOHvis for only one task.

Nevertheless, when it came to the second phase, where participants had experience (from the exploratory phase) in performing tasks, performance was overall better with HyperStorylines for two tasks centered on relationships (**H1**, Find relationships & Characterize relationships): it was faster and perceived easier to use, and it was also less error prone for the Find relationships task. Additionally, we observed that the trends of completion time for both tasks were consistent across tasks variations, with HyperStorylines being faster across the board. Regarding error rate, HyperStorylines was also less error prone in the more complex variations that combined multiple entities

(people+location). It is interesting to note that these tasks were also evaluated in the original PAOHvis paper[7]. We attribute the better performance of HyperStorylines to its simplified and condensed view, as story entities start when their first relationship appears - while with PAOHvis participants needed to both scroll horizontally to follow an entity and identify relationships, and vertically to follow these relationships.

Results for the task that required participants to identify similar entities (in terms of the most common connection) were surprising. Here we were expecting PAOHvis to perform overall better (**H2**), as it displays all the relationships and entities of all types in the same view. In contrast, in HyperStorylines participants had to create the appropriate view, which we expected to be hard to do and time consuming. However, we did not see a difference when considering all tasks variations collectively. When analyzing our results per task variation in the advanced phase, we observe mixed results. On the one hand, PAOHvis is less error prone in the most complex task variation that combines two types of entities (people+locations) over time. But, there is a trend for HyperStorylines to be more correct for the other two tasks variations. We attribute these mixed results to the fact that the benefits of having all entities visible in PAOHvis is only clear in tasks that require many types of entities, while in simpler variations the intense scrolling required to focus on entities of interest overshadows any potential benefits, making HyperStorylines a more effective option.

The results of the task about finding large relationships were also surprising. For this task we were expecting that getting an overview of the whole dataset of relationships and then finding the event with the largest number of entities of one type would be harder with PAOHvis (**H3**). We hypothesized this as in this tool the types of entities are all mixed in the same view, and the length of the lines that represent relationships does not necessarily reflect the number of entities involved but the distance between their position in the list on the left. We expected that these two properties could lead participants to make more errors or take longer to complete the task. However, the only evidence of difference we observed for this task was opposite to our expectations: participants were more prone to making errors with HyperStorylines in the Advanced phase. Nevertheless, when analyzing our results by task variation, we observed this difference only for the most simple task variation that considers one type of entity (people) over time. We attribute this difference to some ineffective strategies adopted by participants in HyperStorylines. For example, two participants chose a view that showed relationships of locations over time, and relied on the number above each relationship to see how many people where involved, instead of searching the longest bar in the view of people related to time. As comparing raw numbers is harder than comparing bar heights, they missed some values and answered incorrectly. Given the observed learning effect of HyperStorylines, it is possible that more time using the tool would have helped participants adopt better strategies for the simple task variations, that could lead to a lower error rate, but this requires further study to confirm.

Overall, our findings are aligned with our initial expectations that becoming an experienced user is more challenging with HyperStorylines in comparison to PAOHVis, as it requires participants to construct appropriate views to answer their questions. Based on participants' comments, this construction was challenging in one out of the four tasks we tested (Large Relationships), leading to more errors when using `HyperStorylines`. However, its condensed visualization design, and its flexibility to build different views of the dataset according to user needs, makes it a powerful tool to explore hypergraphs: participants were faster in two of the four tasks without compromising accuracy (Find Relationships, Characterize Relationships). Furthermore, for one of those two tasks, participants were also more accurate with HyperStorylines. Moreover, HyperStorylines increases user agency, as it provides analysts with control to customize their analysis environment. This is reflected both in the informal feedback provided by our journalists who wanted to progressively drill into the nature of relationships between entities, as well as the increased self-reported confidence and ease of use reported in our comparative study.

Limitations and Future Work

We identify four main limitations in our evaluation of HyperStorylines. The first one is in terms of the size of the datasets used in the study: (i) we did not study how the number of entities (nodes) or number of relationships (hyperedges) impact the use of the respective tools; and (ii) we used datasets that contained only three types of entities (time, people and locations). We deliberately did not evaluate these two scalability issues in order to reduce the factors being studied, but plan to consider them in future studies. In particular, the current design of HyperStorylines allows users to load datasets with any number of *types* of entities (with a minimum of two) and then create partial views of the hypergraph to show relationships containing entities of only *three* types at the same time (those in the horizontal and vertical axes, plus the nested ones). This decision constrains the number of entity types displayed simultaneously and could prevent users from having a complete view of the hypergraph. This design decision is based on the requirements and feedback from journalists, who expressed the need to see less cluttered views of their complex datasets in order to focus on particular stories or types of relationships. Nevertheless, we aim to study in the future how this constraint impacts the exploration of more complex hypergraphs. Apart from using interaction to alternate between entity types, we plan to investigate if a redesign that combines entities of more than one type in one of the axes or in the nested view could be a viable alternative. Related to the last point, we acknowledge that the selection of two initial types of entities to start the exploration might bias the exploration of a hypergraph, a point we aim to explore in future studies. Our goal is to continue improving the design of HyperStorylines to enable the recursive expansion of nested entities for an undetermined number of types of entities, and to evaluate the impact of the dataset size.

The second main limitation of our evaluation is that all our participants were HCI and visualization experts. We feel

they represent a best case when it comes to user expertise and familiarity with new tools. Given their experience in using and designing complex interfaces, they can become experienced users more easily than regular users and detect problems that are independent of the domain of use. Thus, we expect our results in terms of performance to be indicative of what an expert user would attain. More importantly, if these users face difficulties using the tools, we expect these difficulties to hold for other populations (such as journalists). For example, the challenge in constructing views for identifying large relationships is one we need to consider when training our users. We also acknowledge that other populations may require more training to become familiar with the subtleties of the tools, in particular HyperStorylines which is harder to master. Moreover, HyperStorylines is a general purpose visual analysis tool for hypergraphs, even if its design was guided by conversations with journalists. Similarly to other controlled studies, our comparative study provides initial information about how it can be used in specific tasks that are independent of the particular domain, as some have also been used in previous work. Nevertheless, it does not give us a full picture of how HyperStorylines may be used by specific populations in their daily work. A long term evaluation with journalists and other populations, using the tool with their own data, remains future work.

The third main limitation of our evaluation is that our sample consisted of 6 participants only. We believe this sample gives information about the main differences between HyperStorylines and PAOHvis because, as mentioned in previous sections, there is no magic number of participants for user studies[56] and CIs can provide evidence of differences even with two participants[57]. Importantly, the qualitative analysis allows us to understand the reasons behind these differences, as well as to identify the strengths and main problems of HyperStorylines. We believe that the expertise of our participants allowed us to identify the most relevant issues for the evaluated tasks. Nevertheless, we acknowledge that our small sample size may have not unearthed all possible differences. A bigger sample may highlight additional differences between the techniques, albeit ones that are less pronounced or rare. A bigger sample would also give us additional information to reduce the uncertainty of the already identified differences and the size of the calculated CIs. This remains future work.

The fourth main limitation in our evaluation of HyperStorylines is the decision not to allow participants to aggregate by time in HyperStorylines, in order to be comparable to PAOHvis. We hypothesize that for tasks that require users to search inside a time interval, this feature could give an additional advantage as it would allow them to visually filter a subset of relationships to analyze. HyperStorylines already yield more compact timelines and are easier to navigate (as often commented by our participants), and such aggregations will very likely increase this advantage. Nevertheless, activating this aggregation would make the tool incomparable to PAOHvis. It is possible that implementing a time aggregation in PAOHvis can help reduce the confusion mentioned by participants when navigating through a dataset with a large number of relationships (vertical lines) and a long timeline. This would require the redesign of PAOHvis however, and these hypotheses remain to be tested.

For our study we focused on the main visualization without the view of the text documents (news articles) that the relationships come from. Thus our results can be applied to datasets of relationships (hypergraphs) across different domains. Nevertheless, it is likely the visual analytics system where it is embedded, might need to be adapted to each context of use. For example, in our case the raw news-article text is crucial, but we can envision other types of explanations about where this relationship comes from, such as the actual hyperedge, a database table entry or RDF triples, parts of an ontology that generated the relationship, *etc.* We would also like our design to illustrate different types of relationships and data sources that define a relationship. For example, we can consider differentiating relationships between politicians that were extracted from articles from those extracted from Wikipedia, relationships that come from RDF triple stores and ontologies, and even express different levels of uncertainty in relationships.

Finally, there are two additional aspects we would like to further investigate in the future. On the one hand, we would like to include contextual information about entities when possible. For instance, adding information about the topological relationship between the locations involved could allow users to analyze the distance between them and see if it influences the occurrence of certain relationships. On the other hand, we would like to develop means to directly manipulate the hypergraph in the interface. For example, create different aggregation functions in the interface to see their results in real time in HyperStorylines, or merge different hypergraphs from different data sources.

Conclusion

We presented HyperStorylines, a novel generalization of storyline visualization to explore sets of relationships between different types of entities. HyperStorylines was motivated by the needs of investigative journalists and its design was iterated upon in collaboration with them. It provides an interface to build condensed views that allow users to explore the information in a simplified way, and then get more information about relationships on demand. We evaluated our tool with a recently published visualization for hypergraphs, called PAOHvis[7], using four tasks derived from workshops with journalists. Our results show that although HyperStorylines is harder to master at the beginning, it is a powerful and flexible tool to identify and characterize complex relationships in hypergraphs.

Acknowledgements

We would like to thank to the staff from Ouest France who participated in our workshops and gave us feedback of early versions of the tool. We would also like to thank to all the participants of our comparative user study. This work was performed as part of the Inria-funded project lab Knowledge-mediated Content and Data Analytics - The case of data journalism (iCODA).

References

1. Munroe R. XKCD #657: Movie Narrative Charts. http://xkcd.com/657, 2009. Last accessed: 2019-07-27.
2. Ogawa M and Ma KL. Software evolution storylines. In *Proceedings of the 5th International Symposium on Software Visualization*. SOFTVIS '10, Salt Lake City, Utah, USA, 2010: Association for Computing Machinery. ISBN 9781450300285, p. 35–42. DOI:10.1145/1879211.1879219. URL https://doi.org/10.1145/1879211.1879219.
3. Kim NW, Card SK and Heer J. Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces*. AVI '10, Association for Computing Machinery. ISBN 9781450300766, p. 241–248. DOI:10.1145/1842993.1843035. URL https://doi.org/10.1145/1842993.1843035.
4. Reda K, Tantipathananandh C, Johnson A et al. Visualizing the evolution of community structures in dynamic social networks. *Computer Graphics Forum* 2011; 30(3): 1061–1070. DOI:10.1111/j.1467-8659.2011.01955.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01955.x.
5. Silvia S, Etemadpour R, Abbas J et al. Visualizing variation in classical text with force directed storylines. In *Workshop on Visualization for the Digital Humanities, IEEE VIS*.
6. Berge C. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, Amsterdam, 1984.
7. Valdivia P, Buono P, Plaisant C et al. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE Transactions on Visualization and Computer Graphics* 2021; : 1–1DOI:10.1109/TVCG.2019.2933196.
8. Segel E and Heer J. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics* 2010; 16(6): 1139–1148. DOI:10.1109/TVCG.2010.179. URL https://ieeexplore.ieee.org/document/5613452.
9. Kong HK, Liu Z and Karahalios K. Frames and slants in titles of visualizations on controversial topics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18, Montreal QC, Canada, 2018: Association for Computing Machinery. ISBN 9781450356206, p. 1–12. DOI:10.1145/3173574.3174012. URL https://doi.org/10.1145/3173574.3174012.
10. Kong HK, Liu Z and Karahalios K. Trust and recall of information across varying degrees of title-visualization misalignment. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19, New York, NY, USA: Association for Computing Machinery. ISBN 9781450359702, p. 1–13. DOI:10.1145/3290605.3300576. URL https://doi.org/10.1145/3290605.3300576.
11. Riche N, Hurter C, Diakopoulos N et al. *Data-driven Storytelling*. AK Peters Visualization Series, CRC Press/Taylor & Francis Group, Boca Raton, 2018. ISBN 9781138197107. URL https://books.google.fr/books?id=qTOsnQAACAAJ.
12. Brehmer M, Ingram S, Stray J et al. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Transactions on Visualization and Computer Graphics* 2014; 20(12): 2271–2280. DOI:10.1109/TVCG.2014.2346431. URL https://ieeexplore.ieee.org/document/6875900.
13. Glatz E, Mavromatidis S, Ager B et al. Visualizing big network traffic data using frequent pattern mining and hypergraphs. *Computing* 2014; 96(1): 27–38. DOI:10.1007/s00607-013-0282-8. URL https://doi.org/10.1007/s00607-013-0282-8.
14. Eschbach T, Günther W and Becker B. Orthogonal hypergraph drawing for improved visibility. *J Graph Algorithms Appl* 2006; 10(2): 141–157.
15. Kapec P. Visualizing software artifacts using hypergraphs. In *Proceedings of the 26th Spring Conference on Computer Graphics*. SCCG '10, Budmerice, Slovakia, 2010: Association for Computing Machinery. ISBN 9781450305587, p. 27–32. DOI:10.1145/1925059.1925067. URL https://doi.org/10.1145/1925059.1925067.
16. Xie C, Zhong W, Xu W et al. Visual analytics of heterogeneous data using hypergraph learning. *ACM Trans Intell Syst Technol* 2019; 10(1). DOI:10.1145/3200765. URL https://doi.org/10.1145/3200765.
17. Fischer MT, Arya D, Streeb D et al. Visual analytics for temporal hypergraph model exploration. *IEEE Transactions on Visualization and Computer Graphics* 2021; : 1–1DOI:10.1109/TVCG.2020.3030408.
18. Kang H, Plaisant C, Lee B et al. Netlens: Iterative exploration of content-actor network data. *Information Visualization* 2007; 6(1): 18–31. DOI:10.1057/palgrave.ivs.9500143. URL https://doi.org/10.1057/palgrave.ivs.9500143.
19. Ouvrard X, Goff JL and Marchand-Maillet S. Networks of collaborations: Hypergraph modeling and visualisation. *arXiv preprint arXiv:170700115, 2017* ; URL http://arxiv.org/abs/1707.00115. 1707.00115.
20. Beck F, Burch M, Diehl S et al. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum* 2017; 36(1): 133–159. DOI:10.1111/cgf.12791. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12791. https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12791.
21. Bach B, Pietriga E and Fekete JD. Graphdiaries: Animated transitions andtemporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics* 2014; 20(5): 740–754. DOI:10.1109/TVCG.2013.254.
22. Rufiange S and Melançon G. Animatrix: A matrix-based visualization of software evolution. In *2014 Second IEEE Working Conference on Software Visualization*. pp. 137–146. DOI:10.1109/VISSOFT.2014.30.
23. Greilich M, Burch M and Diehl S. Visualizing the evolution of compound digraphs with timearctrees. *Computer Graphics Forum* 2009; 28(3): 975–982. DOI:10.1111/j.1467-8659.2009.01451.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01451.x. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01451.x.
24. Dang TN, Pendar N and Forbes AG. Timearcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum* 2016; 35(3): 61–69. DOI:10.1111/cgf.12882. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12882. https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12882.

25. Arendt DL and Blaha LM. SVEN: informative visual representation of complex dynamic structure. *arXiv preprint arXiv:14126706, 2014* ; URL http://arxiv.org/abs/1412.6706. 1412.6706.
26. Rufiange S and McGuffin MJ. Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics* 2013; 19(12): 2556–2565. DOI:10.1109/TVCG.2013.149.
27. Tversky B, Morrison JB and Betrancourt M. Animation: Can it facilitate? *Int J Hum-Comput Stud* 2002; 57(4): 247–262. DOI:10.1006/ijhc.2002.1017.
28. Lewis TWP. Plotweaver. https://www.metafilter.com/89458/PlotWeaver, 2010. Last accessed: 2019-07-27.
29. Tanahashi Y and Ma KL. Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics* 2012; 18(12): 2679–2688.
30. Liu S, Wu Y, Wei E et al. Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics* 2013; 19(12): 2436–2445. DOI:10.1109/TVCG.2013.196.
31. Tanahashi Y, Hsueh C and Ma KL. An efficient framework for generating storyline visualizations from streaming data. *IEEE Transactions on Visualization and Computer Graphics* 2015; 21(6): 730–742. DOI:10.1109/TVCG.2015.2392771.
32. Liu Y, Lin H, Liang Y et al. An application of optimization method for storyline based on cluster analysis. VINCI '17, Bangkok, Thailand, 2017: Association for Computing Machinery. ISBN 9781450352925, p. 24–28. DOI:10.1145/3105971.3105986. URL https://doi.org/10.1145/3105971.3105986.
33. Tang T, Rubab S, Lai J et al. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Transactions on Visualization and Computer Graphics* 2019; 25(1): 769–778. DOI:10.1109/TVCG.2018.2864899.
34. Tang T, Li R, Wu X et al. Plotthread: Creating expressive storyline visualizations using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics* 2021; 27(2): 294–303. DOI:10.1109/TVCG.2020.3030467.
35. Arendt D and Pirrung M. The "y" of it matters, even for storyline visualization. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. pp. 81–91. DOI:10.1109/VAST.2017.8585487.
36. Ghoniem M, Fekete JD and Castagliola P. A comparison of the readability of graphs using node-link and matrix-based representations. In *IEEE Symposium on Information Visualization*. pp. 17–24. DOI:10.1109/INFVIS.2004.1.
37. Xu K, Rooney C, Passmore P et al. A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics* 2012; 18(12): 2449–2456. DOI:10.1109/TVCG.2012.189.
38. Arafat NA and Bressan S. Hypergraph drawing by force-directed placement. In Benslimane D, Damiani E, Grosky WI et al. (eds.) *Database and Expert Systems Applications*. Cham: Springer International Publishing, 2017. ISBN 978-3-319-64471-4, pp. 387–394. DOI:https://doi.org/10.1007/978-3-319-64471-4_31.
39. Zhao J, Glueck M, Chevalier F et al. Egocentric analysis of dynamic networks with egolines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16, San Jose, California, USA, 2016: Association for Computing Machinery. ISBN 9781450333627, p. 5003–5014. DOI:10.1145/2858036.2858488. URL https://doi.org/10.1145/2858036.2858488.
40. Ouest france. http://www.ouest-france.fr/. Last accessed: 2021-03-21.
41. Görg C, Liu Z, Kihm J et al. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *IEEE Transactions on Visualization and Computer Graphics* 2013; 19(10): 1646–1663. DOI:10.1109/TVCG.2012.324.
42. D3.js. https://d3js.org/. Last accessed: 2021-03-21.
43. D3 layout narrative. https://github.com/abcnews/d3-layout-narrative. Last accessed: 2021-03-21.
44. jlouvain. https://github.com/upphiminn/jLouvain. Last accessed: 2021-03-21.
45. Django. https://www.djangoproject.com/. Last accessed: 2021-03-21.
46. Elasticsearch. https://www.elastic.co/. Last accessed: 2021-03-21.
47. Paohvis. https://gitlab.inria.fr/aviz/paohvis. Last accessed: 2021-03-21.
48. Ahn J, Plaisant C and Shneiderman B. A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics* 2014; 20(3): 365–376. DOI:10.1109/TVCG.2013.238.
49. Alsallakh B, Micallef L, Aigner W et al. The state-of-the-art of set visualization. *Computer Graphics Forum* 2016; 35(1): 234–260. DOI:10.1111/cgf.12722. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12722. https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12722.
50. Kerracher N, Kennedy J and Chalmers K. A task taxonomy for temporal graph visualisation. *IEEE Transactions on Visualization and Computer Graphics* 2015; 21(10): 1160–1172. DOI:10.1109/TVCG.2015.2424889.
51. Andrienko N and Andrienko G. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, Berlin, Heidelberg, 2006. DOI:https://doi.org/10.1007/3-540-31190-4.
52. Jitsi. https://meet.jit.si/. Last accessed: 2021-03-21.
53. Faker python-based library. https://faker.readthedocs.io/en/master/. Last accessed: 2021-03-21.
54. Besançon L, Ynnerman A, Keefe DF et al. The State of the Art of Spatial Interfaces for 3D Visualization. *Computer Graphics Forum* 2021; .
55. Caine K. *Local Standards for Sample Size at CHI*. San Jose, CA, USA: Association for Computing Machinery. ISBN 9781450333627, 2016. p. 981–992. URL https://doi.org/10.1145/2858036.2858498.
56. Bacchetti P. Current sample size conventions: Flaws, harms, and alternatives. *BMC Medicine* 2010; 8(1): 17. DOI:10.1186/1741-7015-8-17. URL https://doi.org/10.1186/1741-7015-8-17.
57. Dragicevic P. Fair statistical communication in HCI. In Robertson J and Kaptein M (eds.) *Modern Statistical Methods for HCI*, chapter 13. Springer, Cham, Switzerland, 2016. pp. 291–330.

58. Cumming G. The new statistics: Why and how. *Psychological Science* 2014; 25(1): 7–29. DOI: 10.1177/0956797613504966. URL https://doi.org/10.1177/0956797613504966.
59. Besançon L and Dragicevic P. The significant difference between p-values and confidence intervals. In *Proceedings of the Conference on l'Interaction Homme-Machine, Poitiers, France, 2017*. pp. 53–62.
60. Besançon L and Dragicevic P. The Continued Prevalence of Dichotomous Inferences at CHI. In *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems*. Glasgow, Scotland Uk, 2019: ACM.
61. Cockburn A, Dragicevic P, Besançon L et al. Threats of a replication crisis in empirical computer science. *Communications of the ACM* 2020; 63(8): 70—79.
62. Cumming G. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge, New York, 2013.
63. Krzywinski M and Altman N. Points of significance: Error bars. *Nature Methods* 2013; 10: 921–922.