Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Visualization and Analysis of Text Documents with Geographic References

Max Franke

Course of Study:	Informatik
Examiner:	Prof. Dr. Thomas Ertl
Supervisor:	Dr. Steffen Koch, M.Sc. Markus John
Commenced:	April 11th, 2018
Completed:	September 24th, 2018

Abstract

In recent years automatic text recognition has gotten more accurate and large amounts of old newspaper articles have been digitised. This allows humanities researchers to use a broader range of computer-aided methods for exploration and sensemaking of this data. Researchers are investigating connections between different publication sources, such as propagation patterns of news. However, they are limited in their pursuit of insight by a lack of tools for exploration and management of such data. Especially for huge datasets, visualisation can greatly improve the search for patterns and anomalies, and help reduce the cognitive load on the user. We create an approach that provides an overview visualisation and exploration capabilities for a collection of newspaper articles, their release dates and publication locations. The approach provides cues on the geographical locations of the visualised data without having to rely on a world map as its core component, freeing up space for other parts of the visualisation. We use brushing and linking throughout the visualisation to provide more context to the user. The dataset can be drilled down into, allowing further exploration, such as filtering by date, location or used words. In an expert review with a researcher in American studies, we evaluate the usefulness of the approach and collect feedback. The general response is that the approach can be useful for researchers. Finally, we provide a discussion and an outlook on future improvements.

Kurzfassung

In den letzten Jahren wurde die automatische Erkennung von gedrucktem Text erheblich verbessert. Dieser Fortschritt ermöglicht die Digitalisierung von historischen Zeitungsartikeln im großen Maßstab. Damit ist es Geisteswissenschaftlern zum ersten Mal möglich, solche Daten computergestützt zu analysieren und damit zu neuen Erkenntnissen zu gelangen. Dabei werden unter anderem Verbindungen zwischen verschiedenen Quellen und die Ausbreitung von Nachrichten erforscht. Ein Mangel an passenden Werkzeugen hindert sie in ihrer Forschung. Vor allem für große Datensätze kann Visualisierung die Erkennung von Mustern und Besonderheiten vereinfachen und die kognitive Belastung für den Benutzer verringern. Wir entwickeln ein Visualisierungskonzept, das für eine Sammlung von Zeitungsartikeln mit Veröffentlichungsdatum und -ort einen Überblick und Möglichkeiten der Exploration liefert. Unser Konzept kann mithilfe von Anhaltspunkten und Kontexthinweisen die geografische Lage der visualisierten Daten darstellen, ohne dabei eine Weltkarte als Kernelement zu nutzen. Damit wird Platz eingespart, in welchem zusätzliche Informationen dargestellt werden können. Wir nutzen in der gesamten Visualisierung Brushing und Linking, um dem Benutzer beim Erforschen der Daten weitere Kontextinformationen zu geben. Der Datensatz kann durch Filtern und Einschränken der Datenmenge weiter exploriert werden, zum Beispiel durch Filtern des Zeitraums, der gezeigten Orte oder erwähnter Worte. In einem Experteninterview mit einer Amerikanistin werten wir die Nützlichkeit unseres Visualisierungskonzepts aus und sammeln Feedback. Die Kernaussage des Interviews ist, dass das Konzept sinnvoll eingesetzt werden kann, um die Beantwortung von Forschungsfragestellungen in der Amerikanistik sinnvoll zu unterstützen. Zum Schluss wird ein Ausblick auf mögliche zukünftige Entwicklungen und Erweiterungen des Ansatzes gegeben.

Contents

List of Figures

1 Introduction and Motivation

In recent years, computer based text digitisation using optical character recognition (OCR) has become very precise. As a result, it is now possible for the first time to digitise large amounts of historical newspaper articles. Historians and researchers of American studies—among others—want to explore those collections of historical newspaper articles, and can for the first time use computer-aided methods to do so on a larger scale. One project doing this is Oceanic Exchanges [45].

Oceanic Exchanges is only one project that aims to analyse '*the global connectedness of* 19^{th} *century newspapers*' [13]. Their data comprises of newspaper articles and their metadata, which include a publication date and location. Their researchers have until now been exploring those newly digitised corpora of newspaper articles in pursuit of answers, and have done so with only minimal tooling. For sensemaking and exploration of large datasets, visualisation can be essential. To help the researchers in their exploration, an interactive visualisation concept should therefore be created. This visualisation concept should be intuitive to use and adhere to Shneiderman's eight golden rules of interface design [34]. Especially should it first provide an overview, and then details on demand. It should also allow easy reversal of actions, and allow the user to be in control. The concept should be able to visualise a corpus of newspaper articles, including its publication location and publication date. We attempt to create such a visualisation concept.

A visualisation concept for document collections with geographical and temporal metadata, if kept general enough, can also be used for other purposes, such as the exploration of information propagation on social media platforms such as Twitter. It could be used to explore spatial and temporal distribution of topics and to gain insights of the sources of information as well as misinformation.

Most visualisation concepts for data with geographical references will use a map as the baseline of the visualisation. This helps the user better understand relationships between data points, following the principle of Tobler's first law of geography: '*Everything is related to everything else, but near things are more related than distant things*' [40, p. 234]. We attempt to create a visualisation concept for data with geographical and temporal references—in particular the publication date and location—which does not use a map as its main element. We try to tackle the problems of letting the users keep their sense of relation between geographical locations, sense of direction and sense of distance.

In this work, we first create a visualisation concept for geographical references within a text that is not based on a map, but instead shows the text itself as central part. We then in multiple iterations create an interactive visualisation concept to view and explore corpora of documents with a publication location and date. We explore a method for visualising geographical relations between objects without using a map by placing objects based on their geographical distance and direction relative to each other. We then compare different projection strategies' intuitiveness for long-distance interrelations.

We succeed in creating a highly integrated visualisation concept that provides an initial overview on the data. The concept allows interaction with all components and utilises brushing and linking to highlight data in different parts of the visualisation. It is possible to drill down into the corpus in order to explore subsets of the data, down to single documents.

We develop the prototype using a dataset provided by Oceanic Exchanges containing 257 newspaper articles from 1883 about the Krakatoa eruption. During the evaluation of the prototype, we perform an expert review with a researcher in American studies from the University of Stuttgart. In that expert review, we gather useful feedback on the needs of the researchers and the shortcomings of the prototype, but conclude that the prototype in itself is a success and can already be of use for the researchers.

We successfully create a visualisation concept for documents with geographical and temporal metadata without centring it around a map. In doing so, we free up a lot of space to be used for other visualisation components. Feedback suggests that the map is not missed as long as sufficient visual cues exist to put the locations into relation. In the end, we provide an outlook on future work which will result in a concept that is even more useful.

2 Fundamentals

This chapter aims to give an overview on the fundamental principles and techniques used in this work. In the first part, we explain the visualisation techniques that we use are explained. Then, we give a brief overview on the principles and difficulties of map projections. Last, we describe the process of hierarchical clustering using dendrograms.

2.1 Visualisation Techniques

The goal of this section is not to give a comprehensive and in-depth list of visualisation techniques, but to give an overview. The visualisation tools used in this work are different sorts of bar charts and word clouds, as well as the technique of brushing and linking. For the visual connection of components, parametric curves based on cubic B-splines are used, which we also briefly introduce.

2.1.1 Bar Charts

(a) Bar chart with baseline. The bars are placed left to right, with a zero baseline at the bottom.

(b) Radial bar chart with baseline. The bars are placed clockwise with the first bar on 12 o'clock.

Figure 2.1: An example for a *bar chart* (a) and *radial bar chart* (b), both showing a sample dataset binned into 11 bins.

Bar charts are used to visualise the values of a small set of data points with a discrete label. A bar chart can thus be interpreted as a discrete density function. Each data point is represented by a rectangular bar, whose length encodes the value associated with the label, as seen in Figure 2.1a. Typically the baselines of the rectangles are aligned.

The discrete labels used for a bar chart can either already be present in the data or created by binning the data. The latter means that a continuous quantitative variable's range is split up into subranges, and the data points of each subrange are aggregated—for example by sum, count or average—to obtain the value for that group. A bar chart that is created by binning is also called a *histogram*.

This work utilises *radial* bar charts as well. In a radial bar chart, each bar has a specific angle such that the bars are distributed around a centre point, as seen in Figure 2.1b. The base line is represented by a circle around that centre. The radial bar charts used throughout this work are used primarily to show data with an angular component, such as the directional histogram introduced in Section 4.2.3. The second use is for appending a bar chart to a round component of the visualisation, where a regular bar chart would seem out of place. We use this for surrounding our round map insets—introduced in Section 4.2.4—with date histograms.

2.1.2 B-Splines

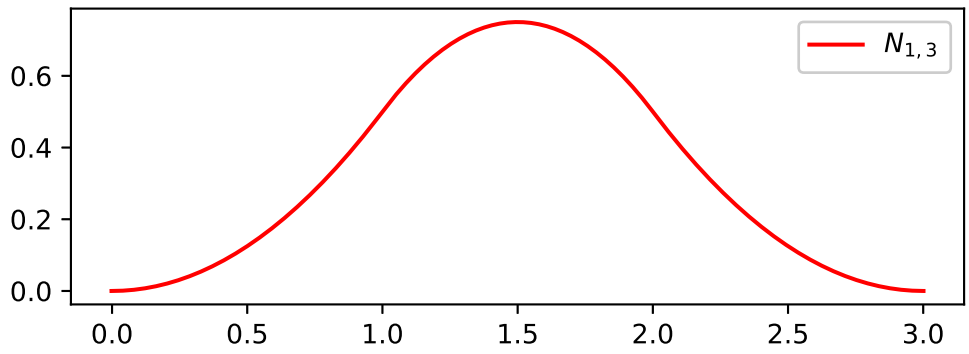


Figure 2.2: A cubic B-Spline basis function $N_{1,3}$ for the knot vector $(0, 1, 2, 3)$.

The term *B-spline* is short for '*basis spline*'. This section is based on the formulas and explanations provided by the University of Cambridge [47] and from de Boor [5].

A B-spline S of order n is defined as a piecewise polynomial function of degree $n-1$. It is defined by k control points $p_i \in \mathbb{R}$ and $n+k$ knots $t_i \in \mathbb{R}$. The knots define the domains of the basis functions, and the control points weigh the n^{th} order basis functions, resulting in the spline. The basis functions of order n are recursively defined via the basis functions of degree $n-1$ as follows:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,n}(t) = \frac{t - t_i}{t_{i+n-1} - t_i} N_{i,n-1}(t) + \frac{t_{i+n} - t}{t_{i+n} - t_{i+1}} N_{i+1,n-1}(t)$$

The spline S is then defined as the weighted sum of its n^{th} degree basis functions:

$$S(t) = \sum_{i=1}^{k+1} N_{i,n}(t) \cdot p_i$$

Figure 2.2 shows a cubic basis function for a knot vector $(0, 1, 2, 3)$. In this work, we use parametric curves based on cubic B-splines. Those parametric curves are constructed by creating two separate splines S_x, S_y using the x and y components of the control points over an independent variable t, and then evaluating both splines for the same t to obtain the points on the parametric curve C:

$$C : \mathbb{R} \to \mathbb{R}^2$$
$$C : t \mapsto \Big(S_x(t),\ S_y(t)\Big)$$

Using parametric splines allows us to define start and end points for lines as well as control points. The result is smoothly curved connections between components where we can control the start and end angles of the lines. For instance, in Section 4.1, we use this to start the lines horizontally. Another benefit of splines is that they are mathematically simple and therefore cheap to calculate, which means they do not need much computing power to be drawn. Splines can also be directly specified in SVG[1] by their control points, meaning we do not have to worry about the details of constructing them.

2.1.3 Word Clouds

Word clouds are a visualisation technique used to give an indication on the importance or frequency of words in a text or collection of texts. A subset of the most important words from the text collection are displayed with the word's importance encoded in its size. The result is a cloud of words with different sizes, where the larger words attract the most attention.

Calculating the importance of the words requires some thought. First, words that are too frequent and carry little information are removed. Those words are called *stop words* and can be considered noise in the visualisation data. Common stop words are 'the', 'a' and 'and'.

Second, there are are different measures that can be applied to a collection of words to determine their importance. Let

$$\mathcal{D} = \{d_i\}_{i=1}^{I}$$

be a collection of I documents $d_i, 1 \leq i \leq I$ and

$$d \in \mathcal{D},\ \ d = \left\{w_j\right\}_{j=1}^{J}$$

be a document, a collection of J words $w_j, 1 \leq j \leq J$. Then the following three measures can be defined [29, pp. 68 sqq.]:

- **Term frequency:**
  Term frequency tf is the number of times the word appears in the document collection $\mathcal{D}$:

$$tf : w \mapsto \sum_{i=1}^{|\mathcal{D}|} \left|\left\{w' \;\middle|\; w' \in d_i \wedge w' = w\right\}\right|$$

[1] Scalable Vector Graphics — A vector graphics format that can be rendered by modern web browsers and is used for complex graphics on web pages.

- **Document frequency:**
  Document frequency *df* counts the amount of documents d in the collection $\mathcal{D}$ the word appears in. Each document is only counted once:

$$df : w \mapsto \left|\{d \in \mathcal{D} \mid w \in d\}\right|$$

- **Term frequency - inverse document frequency:**
  tf-idf attempts to filter out terms that appear in all or nearly all documents in the collection, instead giving more importance to terms only appearing in few documents, but often there. *tf-idf* is calculated by multiplying the term frequency *tf* of a word with the *inverse document frequency* of the word. The inverse document frequency *idf* is calculated by taking the natural logarithm of the inverse relative document frequency:

$$idf : w \mapsto \ln \frac{|\mathcal{D}|}{df(w)}$$

  leading to the formula for *tf-idf*:

$$\textit{tf-idf} : w \mapsto tf(w) \cdot idf(w)$$

It should be noted that for the words in a word cloud, the importance of a word should not be encoded in the font size, but instead in the *area*, leading to a approximate mapping from the importance or weight ω to the font size ϕ of

$$\phi \sim \sqrt{\omega}$$

Using a linear mapping to the font size would result in a large *lie factor* [41] as users would intuitively perceive the area as the visual variable encoding the importance.

Word clouds are a visualisation technique that removes information. For instance, the text structure is completely lost, which also takes away possible negations, irony, or quoted text. Harris argues that word clouds can be very misleading because they '*support only the crudest sorts of textual analysis*' [23, p. 2]. In conclusion, word clouds should not be used as a core visualisation technique, but at most as a tool to give a brief overview on the topics of a text or collection of texts.

2.1.4 Brushing and Linking

Brushing and linking are interaction techniques used to give the user of a visualisation a relation between components and data points. Both are used in combination to relate data points in different parts of a visualisation. Buja [9] presents brushing and linking in combination with *multiple coordinated views*, where each view only covers partial information on the data, to relate data points between singular views of the visualisation.

Brushing is the interaction of selecting a subset of the data, for example by hovering over a data point or selecting all data points in an area by clicking and dragging the mouse. *Linking* means that this subset of data points is then highlighted not only where it was selected, but also in every other part of the visualisation it appears in.

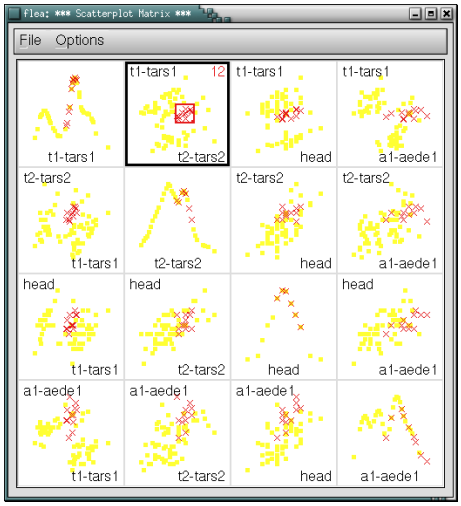
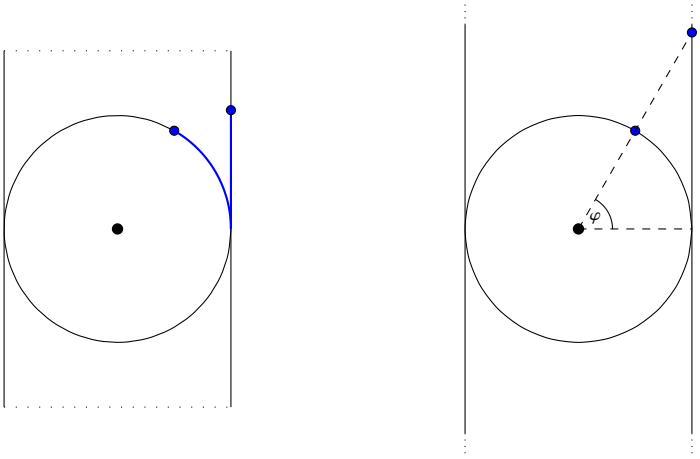


Figure 2.3: An example of brushing and linking in a scatter plot matrix, in the visualisation tool *GGobi* [38]. A subset of data points are brushed in the selected subplot and linked in the others by using a different colour.
Image source: https://infovis-wiki.net/w/images/6/61/Ggobi04brushlink.png, retrieved on July 21st, 2018.

One example might be a scatter plot matrix, where each subplot shows the data from a different perspective, as shown in Figure 2.3. Highlighting a data point or selection of points in all subplots when the user selects them in one plot is a good example of brushing and linking. It enables the user to interactively explore part of the data. Keim et al. [26] argue that linking and brushing in multiple views conveys more information to the user than just showing him the views. Linking and brushing are a good example of the *relate* task listed by Shneiderman [33].

2.2 Map Projection

Projecting the earth's close-to-ellipsoidal surface onto a two-dimensional plane in a sensible manner poses several problems. Regardless of the method, the flattening introduces a number of possible distortions, such as distortion of distance, direction, area, shape, or even breaks in the map area. Over the centuries, countless map projections have been introduced in an attempt to minimise the distortion, however minimising one aspect usually forces another. In the following, we will introduce two map projections which are relevant for this work. For further reading on map projections, we refer to John P. Snyder's *Flattening the Earth* [36], which provides a comprehensive list of projections from the early beginnings of cartography until the twentieth century, highlighting their individual strengths and weaknesses.



(a) Equirectangular projection. **(b)** Cylindrical projection.

Figure 2.4: Equirectangular (a) and cylindrical (b) projections. Both projections are based on a cylinder. For equirectangular projection, latitude and longitude are interpreted as points on a cylinder instead of a sphere or ellipsoid. For cylindrical projection, the location on the sphere or ellipsoid is projected from the surface onto the cylinder. The cylinders are then unrolled to obtain a flat surface.

2.2.1 Equirectangular Projection

Equirectangular or Plate Carrée[2] projection provides a very simple way to project coordinates onto a flat surface. Let λ be the longitude, φ the latitude, φ_1 the standard parallels[3], λ_0 the central meridian, and x and y the coordinates on the flat map. Let R be the radius of the earth's sphere[4]. Then the projection is calculated as follows:

$$x = R \cdot (\lambda - \lambda_0) \cos \varphi_1 \qquad y = R \cdot \left(\varphi - \varphi_1\right)$$

For the Plate Carrée projection, which uses $\varphi_1 = 0$, with the central meridian being the Greenwich meridian, the calculation simplifies further to

$$x = R \cdot \lambda \qquad y = R \cdot \varphi$$

This calculation equates to converting the sphere to a cylinder by stretching the surface towards the poles, and then unrolling the cylinder into a flat surface. A points distance from the equator is thus equal on the sphere and in the projection, as depicted in Figure 2.4a. The lateral distances get more distorted the closer one comes to the poles. The poles themselves are projected from a point on the sphere to a line in the projection that is as long as the equator.

[2] Plate Carrée projection is a special case of equirectangular projection where the equator is the *standard parallel*. The standard parallel is the latitude at which the scale of the projection is true.

[3] See Footnote 2.

[4] This is the simplified version. Modelling the earth as a ellipsoid modifies the formulas.

2.2.2 Mercator Projection

Mercator projection aims to project a straight line on the sphere onto a straight line in the projection. Such a line is called a *loxodrome*. For such a line, a change in longitude is always proportional to the distance between two arbitrary meridians:

$$\delta\lambda \sim \frac{1}{\cos\varphi} = \sec\varphi$$
$$x = R \cdot \lambda$$

The latitude is projected as shown for the cylindrical projection in Figure 2.4b. While the cylindrical projection would then simply use the tangent of the latitude to calculate the position in the projection, Mercator projection scales this value by the longitudinal scaling factor to keep the loxodrome property:

$$\begin{aligned} y &= R \cdot \int_0^{\varphi} \sec\varphi \, \mathrm{d}\varphi \\ &= R \cdot \ln\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right) \end{aligned}$$

Like the equirectangular projection, Mercator projection distorts areas and distances towards the poles. Even more so, the poles can never be part of the projected map, as the integral becomes unbound towards ±90°. Nevertheless, according to Snyder [36, pp. 45 sqq.], Mercator projection became popular in nautical circles because it enabled the sailors to plot straight courses—loxodromes—on the map. Gudmundsson and Alerstam even argue that because of its ubiquity in nautical circles, Mercator projection permanently shaped our mental image of the world: '*The indispensability of the Mercator projection for navigational charts and its long and widespread use has contributed to make a lasting impact on our view of the world. It is important to remember that [. . .] it is true to neither distance nor area*' [21, p. 601].

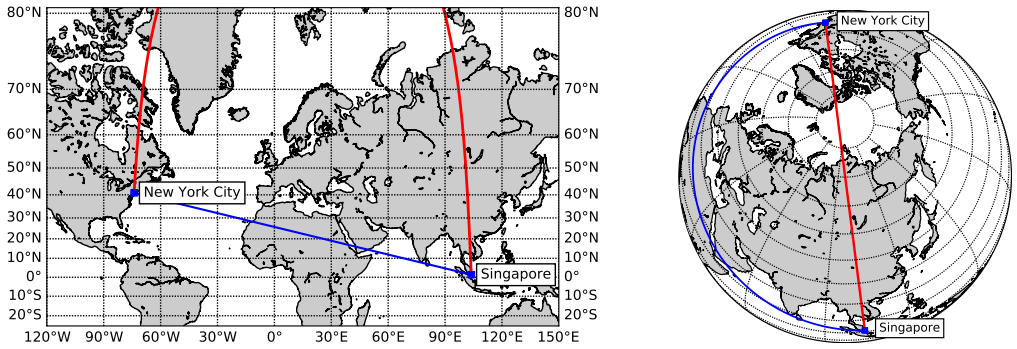
2.2.3 The Great Circle Distance and Direction

In order to calculate distance and direction of two points on a perfect sphere, the great circle can be used. The great circle—or *orthodrome*—is a circle around the sphere whose centre is the centre of the sphere[5]. The route along the shorter part of the great circle on which two points lie is the shortest path between these points following the surface of the sphere.

The distance d of two points $(\varphi_1, \lambda_1), (\varphi_2, \lambda_2)$, where λ is the longitude and φ is the latitude, can be calculated as follows according to Simmons and Gore [35]:

$$\Delta\sigma = \arccos\left(\sin\varphi_1 \cdot \sin\varphi_2 + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \cos\left(\Delta\lambda\right)\right)$$
$$d = R \cdot \Delta\sigma$$

[5] Another way of defining a great circle is to say that a great circle is the intersection between a sphere and any plane that contains the centre of the sphere. Examples for great circles on Earth are the equator or the Greenwich meridian together with the 180$^{\text{th}}$ meridian.



(a) Mercator projection.

(b) Orthogonal projection on a sphere.

Figure 2.5: The great circle arc depicting the shortest path between Singapore and New York City. On the left, the great circle is shown using Mercator projection. As can be seen, the intuitive shortest path (blue) is very different from the actual shortest path (red). On the right, the great circle arc is plotted onto a sphere, which models Earth with high accuracy. Here, the great circle arc is actually a straight line along the surface of the sphere.

The direction or *azimuth* of the great circle arc changes over the course of the arc[6]. The start azimuth α_1 and end azimuth α_2 can be calculated as follows:

$$\tan \alpha_1 = \frac{\sin(\Delta\lambda)}{\cos\varphi_1 \cdot \tan\varphi_2 - \sin\varphi_1 \cdot \cos(\Delta\lambda)}$$
$$\tan \alpha_2 = \frac{\sin(\Delta\lambda)}{-\cos\varphi_2 \cdot \tan\varphi_1 + \sin\varphi_2 \cdot \cos(\Delta\lambda)}$$

The great circle arc connecting two places on earth can often look vastly different from what one would expect given our world view. As can be seen in Figure 2.5, the great circle arc (red) between Singapore and New York City very nearly crosses the North Pole and cannot even be completely drawn using a Mercator projection. The fact that it still represents the shortest path between those two points shows clearly that our intuitive understanding of distance and direction on a sphere can sometimes be off by a lot.

2.3 Hierarchical Clustering and Dendrograms

Clustering is the process of grouping data points into *clusters* such that all points in one cluster are closer to each other than a specified threshold. The goal of clustering is to sort the data into categories programmatically. The choice of the similarity function or distance function and the proper threshold is essential.

Hierarchical clustering is a form of clustering that has multiple layers with growing threshold. The first set of clusters are again clustered using a higher threshold. This is repeated either a set number of times or until only one big cluster remains.

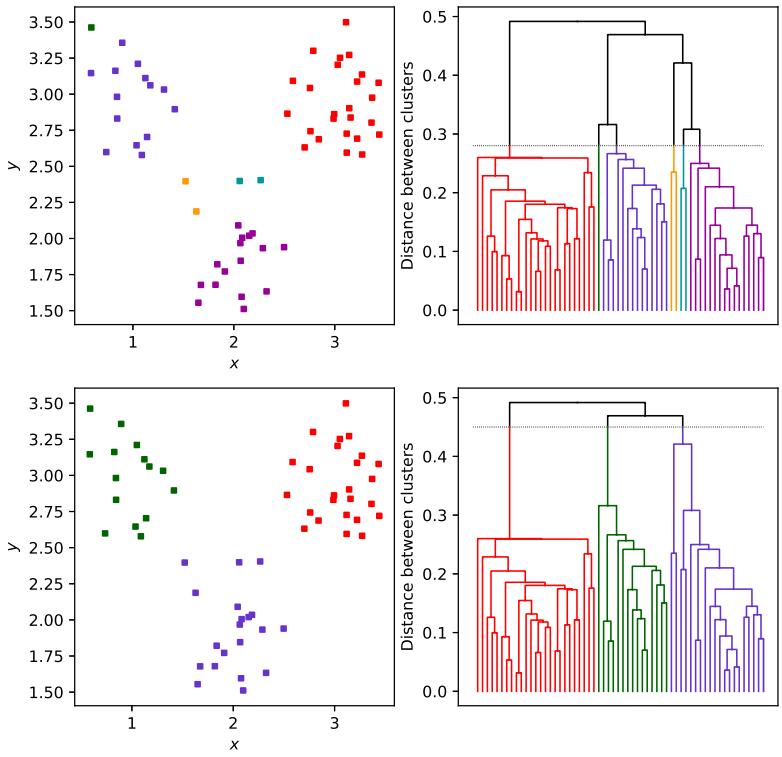


Figure 2.6: The right side shows a dendrogram created from the two-dimensional data shown in the scatter plots in the left, using Euclidean distance. Top and bottom show two different thresholds for cut-off and the resulting clusters.

A *dendrogram* is a graph structure that can be used to construct hierarchical clustering. It is a binary tree where the leaf nodes are the data points and the intermediate nodes notate the distance between the two child nodes. The algorithm, as specified in Alg. 2.1, works by creating a new cluster containing the two closest clusters in each step. Each cluster also gets an attribute containing the distance between its two children.

From this representation, clusters can be formed using a threshold by cutting off the tree's subtrees where the distance is larger than the threshold. The resulting subtrees represent the clusters, with the leaf nodes being the clusters' data points. Figure 2.6 shows a dendrogram that is cut off using two different thresholds, and the resulting clusters. The visualisation shows a popular method of drawing dendrograms, where the distance between the child clusters is encoded in the height of the subtree.

[6]Except for the great circle that is the equator. Here, the azimuth stays at 90° everywhere.

Algorithm 2.1 Creating a Dendrogram

Input: D Array of data points
$\text{dist} : C \times C \to \mathbb{R}$ Distance function
Output: C Binary tree root.

$C = D$ // Each data point is a one-element cluster.
while $|C| \neq 1$ **do** // While not all nodes clustered.
  $(i, j) \leftarrow \text{argmin}_{i,j} \left(\text{dist}(c_i, c_j),\ \ c_i, c_j \in C,\ i \neq j \right)$ // Select two closest clusters.
  $c_1 \leftarrow c_i \in C$
  $c_2 \leftarrow c_j \in C$
  $C \leftarrow C \setminus \{c_i, c_j\}$ // Remove clusters from list.
  $c' \leftarrow \left(\text{dist}\left(c_i, c_j\right), c_i, c_j \right)$ // Create intermediate node.
  $C \leftarrow C \cup \{c'\}$
end while
return C

3 Related Works

3.1 Methods for Visualisation of Off-Screen Elements

Attempting to fit all parts of a dataset on the screen in a visualisation creates multiple problems. First, the amount of data shown may lead to single data points becoming very small. This is even more so the case if the proper scale and distance is kept without distortion. In particular, this means that the ratio of two distances or areas in the visualisation should be the same as their ratio in the real world. For geo-located data points, this cannot always be guaranteed. As demonstrated in Section 2.2, keeping a map undistorted when projecting it to a two-dimensional surface is non-trivial, if not impossible.

Keeping the distances intact in most datasets also leads to large areas of unused space between areas of higher density. Second, exploration of the dataset is made very hard, especially with a varying information density. While the overview exists in the full view, zooming and panning to discern details in the data will disrupt the user's sense of orientation.

In the following, we list a few approaches used in other works to either show a contextual overview alongside the core visualisation or provide cues in the visualisation that help the user putting the visible area in context.

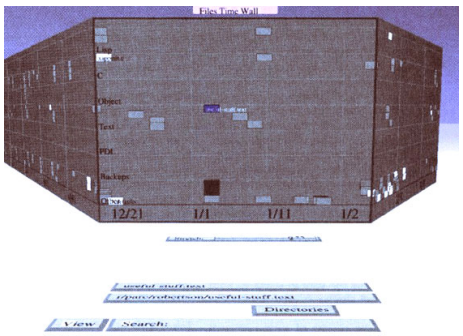
3.1.1 Varying Density and Distortion Approaches

Baudisch et al. [2, 3] propose a solution for these problems by introducing a technique by which the centre part of the display area has a higher pixel density. This technique utilises the way the human vision focuses largely on the centre of the visual field, while in the peripheral only coarse shapes and colours are registered. This technique shows a way to retain some overview, but it does not scale well with greater zoom levels.

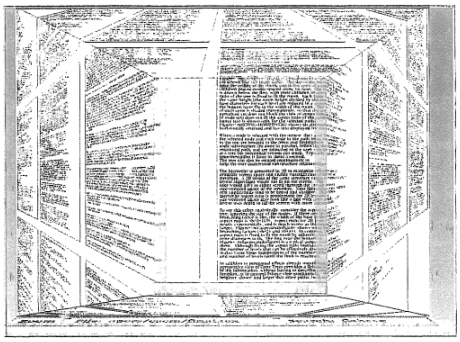
3.1.2 Distortion-Based Approaches

Cockburn et al. [12] provide a collection of interfaces that provide a context alongside the focused data. They group interfaces that need neither a spatial nor a temporal (zooming) separation between the focus and the context as Focus+Context. Such interfaces are usually implemented by some kind of distortion.

Furnas [18] introduces the concept of varying levels of detail in different parts of the visualisation. The level of detail would be higher in areas with a higher degree of interest. Furnas calls such a view a *fish eye view* as it is distorted like the visual field of a fish. Mackinlay et al. [28] use this concept for their *perspective wall*, which shows a part of the information in the centre and the



(a) The perspective wall [28]. Image source: https://infovis-wiki.net/w/images/7/7f/Perspective_wall.jpg (Obtained July 24th, 2018).



(b) *DocumentLens*. Image from the original paper [31, p. 6].

Figure 3.1: The perspective wall by Mackinlay et al. (a) and the DocumentLens by Robertson and Mackinlay (b). Both visualisations utilise a fish eye view to give the user an overview around a detailed part.

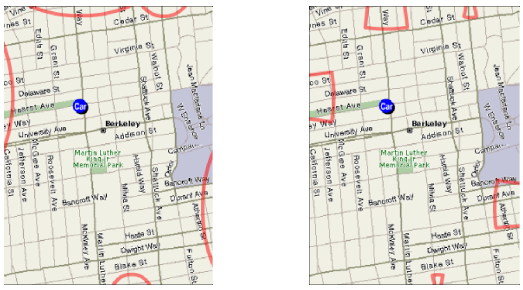
surrounding data in perspective. Figure 3.1a shows the perspective wall. Robertson and Mackinlay continue this concept with the *DocumentLens* [31] (Figure 3.1b). Leung and Apperley [27] give an overview on distortion-oriented visualisations as well.

All those approaches do not necessarily visualise off-screen content, but instead vary the level of detail towards the edges of the display area, with the effect that the whole domain or a large part of it is directly visible on the screen, with a part of it in greater detail. This allows the user to retain the overview and a feeling for which part of the visualisation is currently visible.

3.1.3 Cue-Based Approaches

Another technique, which Cockburn et al. [12] classify as *cues*, show off-screen elements as decorations at the edges of the visible part of the data. Some implementations of that technique are shown in the following. These are used both for visualising geographical data, where the layout is already given and off-screen locations are cued, and for abstract graph data (e.g. UML diagrams), where off-screen parts of the graph are cued.

Baudisch and Rosenholtz [4] introduce *Halo*, a technique to indicate the direction and distance of off-screen elements. *Halo* draws ring segments (cf. Figure 3.2a) at the screen edges in such a way that the centre of the ring would be the off-screen location it represents. The user can then deduce the approximate location from the position and curvature of the ring segment. Gustafson et al. [22] improve this technique with *Wedge*, replacing the ring segments with wedge segments (cf. Figure 3.2b). This improves the technique especially in cases where *Halo*'s ring segments would overlap. Both visualisations are helpful for giving the user an indication of the location and distance of off-screen data points. However, they lack the possibility to encode much more than the location into the indicators.



(a) *Halo*. **(b)** *Wedge*.

Figure 3.2: Comparison of *Halo* (a) and *Wedge* (b). Both images are taken from *Wedge: Clutter-Free Visualization of Off-Screen Locations* [22].

Frisch and Dachselt [17] study a method to explore UML diagrams, allowing the user to zoom in while retaining the overview. In their prototype, off-screen UML elements that have links to elements within the screen space are hinted at via so called proxy elements at the margin of the screen. Several proxies can represent the same off-screen element. The approach of Frisch and Dachselt makes it possible to add some additional information into the indicators. In their prototype, they differentiate different types of UML element's proxies by their shape and label.

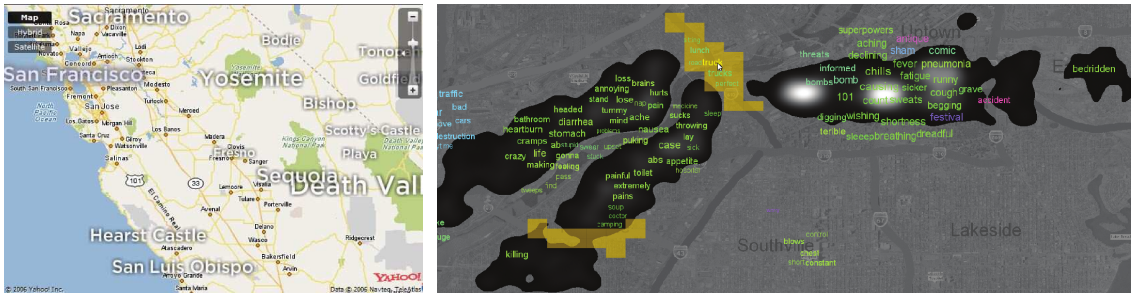
Ghani et al. [19] use a similar method in combination with maps. Their approach for navigating graphs with geographical references shows off-screen nodes in an inset map at the edge of the screen. The positioning of the insets depends on the geographical position of the contained node in relation to the visible map area. This approach makes it possible to view a graph with fixed positions of the nodes in detail while keeping the overview.

Brodkorb et al. [8] further improve this method by simplifying geo-located graphs. They move remote parts of the graph into insets and perform local rearrangement of nodes to reduce overdrawing and edge-crossing. Their approach makes exploration of geo-located graphs easier and more intuitive. The overview is kept while dense areas are magnified and empty space is reduced.

3.2 Visualisation of Geo-Located Data

Multiple approaches exist to visualise geo-located data. Most approaches use a map as their central element. This is the most intuitive approach and gives the user a means to relate the data to their own view of the world. However, many datasets consist of areas with high information density separated by areas with very low information density and sometimes no data at all. Examples are population statistics, which only have data points in populated areas, or water quality measurements, which only have data points along rivers and lakes, but not in the countryside in between. In a map-centred visualisation without distortion, this leads to crowding as described by Rosenholtz et al [32] and unused areas of the canvas.

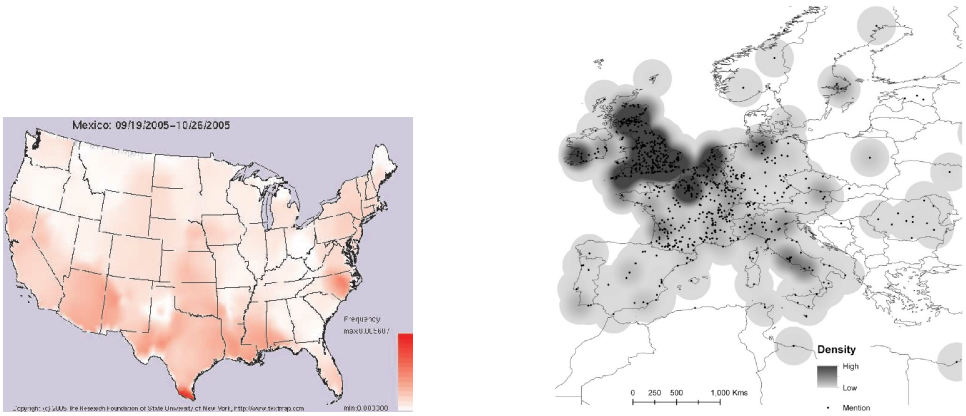
WorldExplorer [1] (Figure 3.3a) draws tags into a map at their most prominent location. Other tags for that location are proposed as secondary tags when hovering over the tag. *ScatterBlogs* [6] (Figure 3.3b) uses a very similar technique to visualise prominent tags for different locations. Jaffe et al [25] employ a similar technique, clustering photographs at referenced geo-locations. Photographs



(a) *WorldExplorer*. **(b)** *ScatterBlogs*.

Figure 3.3: *WorldExplorer* (a) and *ScatterBlogs* (b). Both images are taken from the original papers [1, 6].

with a similar motive and geo-location are merged into one representative photograph at the location. These approaches are suited to give the user a good first overview of the topics or tags that are most important in certain locations. However, large areas of the visualisations are unused, although for the Lakeside district in Figure 3.3b this might be a useful part for the visualisation, as the task that had to be solved with the tool was to find the ground zero location of an epidemic.



(a) The visualisation by Mehler et al. [30, p. 766] for the distribution of the word 'Mexico' in news sources across the United States.

(b) The visualisation by Gregory and Hardie [20, p. 306] of place-names from the Lancaster Newsbook Corpus.

Figure 3.4: The density based topic visualisations by Mehler et al. [30] and Gregory and Hardie [20]. Both images are taken from the original papers.

Mehler et al. [30] (Figure 3.4a) use density estimation via a heat map to show the distribution of a single topic across a map. Gregory and Hardie [20] (Figure 3.4b) also use density estimation in their approach, but additionally draw the individual data points into their geo-locations. Those two approaches can give the user an overview on the distribution of one topic. In combination with linking and brushing, they can be useful for exploration of datasets. By themselves the approaches are very limited, as only one topic can be visualised at a time, and the differences between two topics can only be explored by either spacial separation—side by side—or by temporal separation—animation.

3.3 Visualisations with Radial Layout or Components

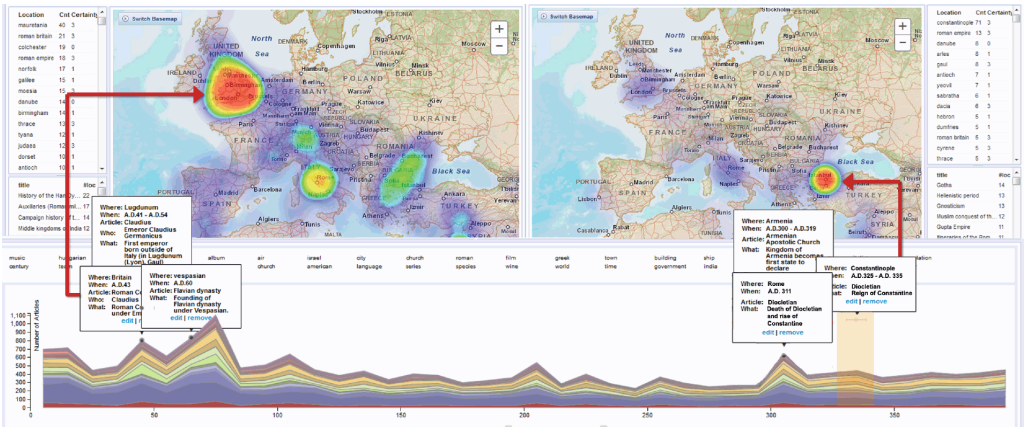
This section gives a brief overview on visualisation techniques that use a radial layout. Such layouts are often space-efficient, aesthetically pleasing and—when combined with interaction techniques such as brushing and linking, and animations—intuitive. Draper et al. [14] provide an overview of radial methods and name their strengths and weaknesses.

VAiRoma [11] is a visual analytics system designed to explore places, times and events in Roman history. *VAiRoma* includes multiple different views, including a map view, a time line and two types of topic tree views: As textual tree and as circular view. The map view (Figure 3.5a) can incorporate pins, labels and a heat map overlay to display hot spots. The circular topic view (Figure 3.5b) shows topics in a radial tree. The topics are also visualised as circles outside the radial tree, showing their own weight. A topic's keywords are visualised around the topic circles. The centre of the radial tree shows either the selected time period or a tag cloud of the topic keywords. The latter approach is similar to our word cloud centred view as described in Sections 4.2.3 to 4.2.5. *VAiRoma* supports linking, brushing and other interaction techniques to facilitate exploration of the data.

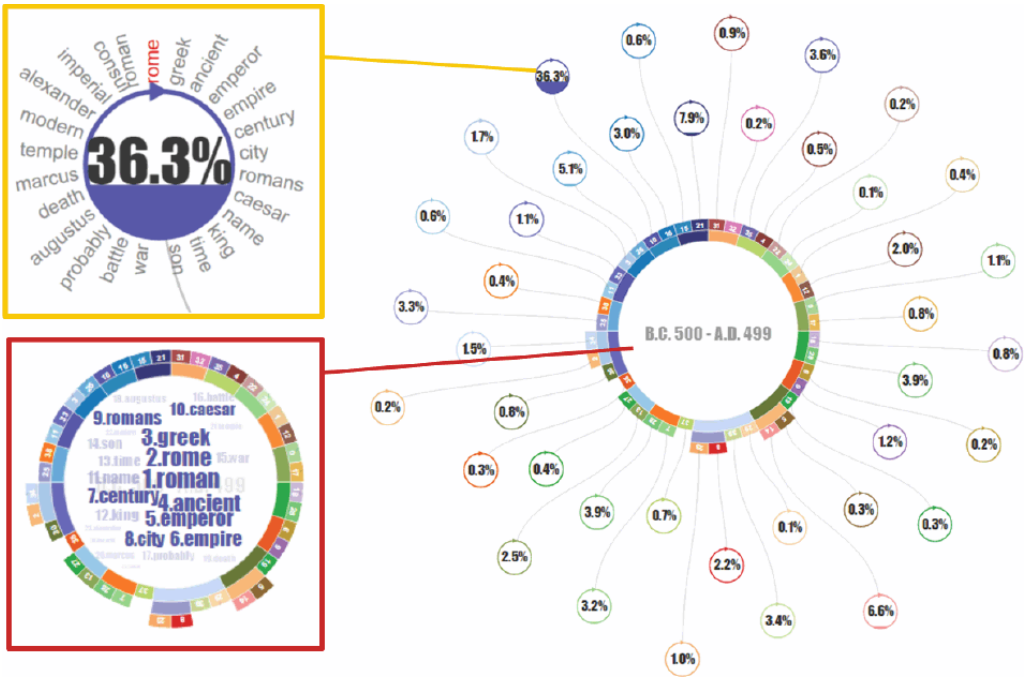
Whisper [10] (Figure 3.6) is a visualisation tool to monitor the spreading of information via Twitter. It uses the distance from the centre of the visualisation, where the topic disc is shown, to encode time passed since the original tweet was sent. Retweets are marked as ticks on lines originating from the topic disc towards a user group at the edge of the circle, positioned on the isoline representing the time difference between the original tweet and the retweet. The user groups can be countries, as in Figure 3.6a, or states (Figure 3.6b). In the case of countries in Figure 3.6a, which the authors describe as '*longitude layout*' [10, p. 2654], the angle towards the user group nodes encodes their geographical longitude. To make this more tangible to the user, a pole-centred Lambert azimuthal projection is drawn under the time scale to give a sense of direction. In Figure 3.6b, '*circular layout*' [10, p. 2654] is used, which positions the nodes equally-spaced around the circle. *Whisper* gives the user a tool to analyse the propagation of news—in this case tweets—and even provides a sense of sentiment of the tweets. However, the centre of the background map is arbitrary and only provides a slight indication of direction. Nodes are not drawn in the correct position on the map, and that area is actually reserved for the time difference encoding.

Drocourt et al. [15] (Figure 3.7a) propose a radial visualisation where spatial distance is mapped to the angle and temporal distance mapped to the distance from the centre. Their approach expects the data points to have an order, and the angle differences between two points are only dependent on the two data points' distance. This approach worked well for the data points they were visualising, which were located along the coast line of an island, specifically Greenland. The approach would not work as well for points with no intuitive way of reduction from 2D Cartesian coordinates to 1D angular coordinates. Points in time are represented by concentric circles, where the outer circles would represent more current data. The approach facilitates the viewing and comparison of multiple data points' trend over time, but fails when no intuitive order can be established between data points.

Stasko and Zhang [37] (Figure 3.8) propose three visualisation and interaction techniques that give a focus+context approach to hierarchical radial visualisation layouts: *Angular Detail* (Figure 3.8a) shows the currently focused area as larger-than-scale slice and keeps the parent hierarchy as overview. *Outside Detail* (Figure 3.8b) shows the focused data as outer ring and the overview in the centre of the ring. *Inside Detail* (Figure 3.8c) shows the focused data in the centre and the parent hierarchy

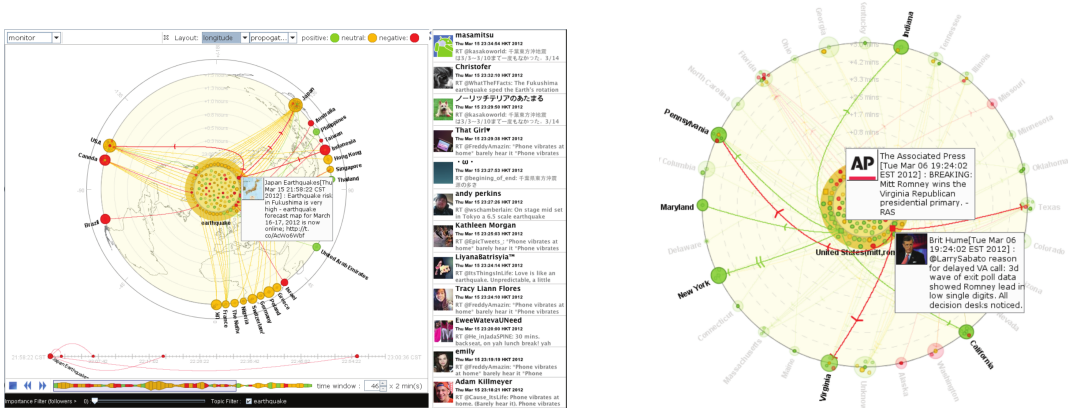


(a) Comparison view between two time periods for the topic ‘war’.



(b) The circular topic view.

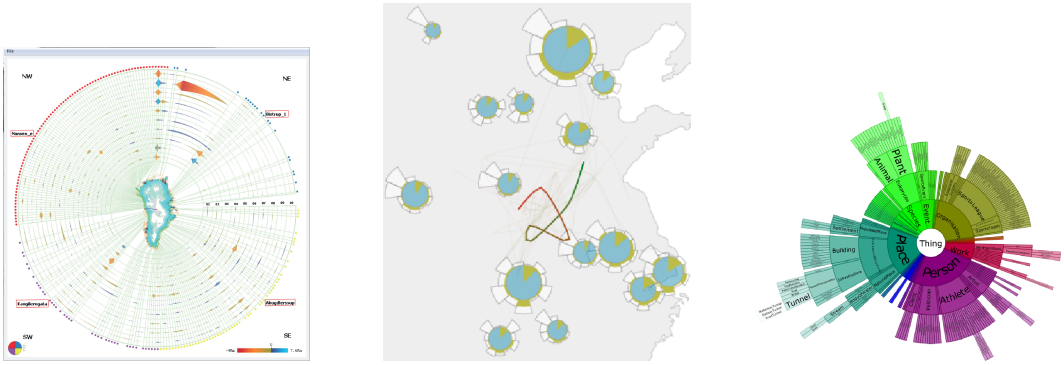
Figure 3.5: The *VAiRoma* [11] visualisation tool. In the map view (a), the user can for instance see a heat map overlay for hot spots of topics. In the shown view, two time periods are compared side by side. The circular topic view (b) shows a hierarchical distribution of topics in the sunburst in the centre. The text in the centre can be the current range or the topic keywords (red rectangle). The circles around the outside each represent a topic along with its weight, which is shown enlarged within the yellow rectangle.
All images are taken from the original paper (downloaded in better quality from https://ieeexplore.ieee.org/abstract/document/7192676/ on August 13th, 2018).



(a) The user interface of *Whisper*. Retweets are bundled by country, and the locations are projected from the north pole in 'longitude layout' [10, p. 2654].

(b) Only tweets from the United States are shown. The retweets are bundled by state in 'circular layout' [10, p. 2654].

Figure 3.6: The user interface of *Whisper* [10] on the left shows in its central visualisation component a collection of tweets in the centre. The outward concentric circles represent time passed since the original tweet, and marks along the lines represent retweets of the original tweet. The retweets are bundled to *user groups*, which here are countries. Below the time circles, a Lambert azimuthal equal-area projection from the north pole is drawn to give a reference to the locations of the retweets. In the second view (b) the user groups are the different states of the USA, and no background map is drawn. Both images are taken from the original paper.

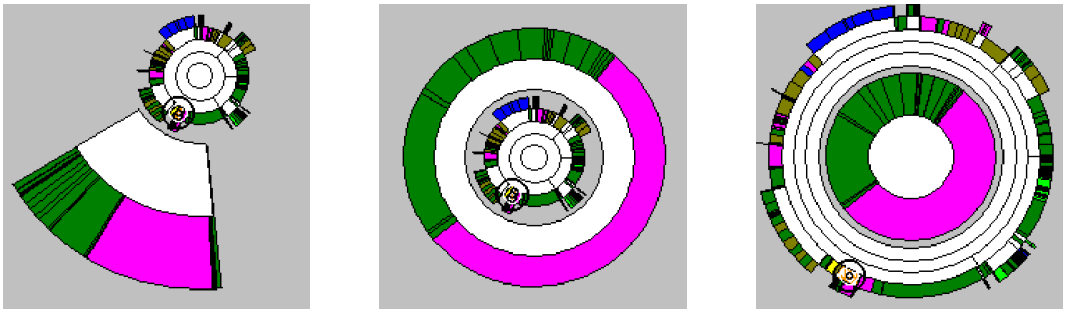


(a) Radial design by Drocourt et al. [15].

(b) Topic trajectory visualisation by He and Chen [15].

(c) Radial tree visualisation in *ViDaX* [16].

Figure 3.7: Visualisations with a radial design. Images are taken from the original papers [15, 16, 24].



(a) Angular Detail. **(b)** Outside Detail. **(c)** Inside Detail.

Figure 3.8: The three focus+context visualisation and interaction techniques introduced by Stasko and Zhang [37]. Angular Detail (a) lets the user drill into one subtree of the hierarchy. Outside (b) and Inside (c) Detail show the context inside resp. outside the currently focused data. The images are taken from the original paper.

in a ring outside. All three techniques give the user a good reference from the currently viewed data to the rest of the dataset. Combined with smooth animations, the reference is never lost. However, the approach only works for strictly hierarchical data that can be viewed in a radial tree view. *ViDaX* [16] also uses a radial tree layout, as shown in Figure 3.7c.

He and Chen [24] show geographical trajectories of topics. Local dense areas are clustered, and the clusters represented as radial glyphs, where the angle represents the time in a radial histogram. Their visualisation, as shown in Figure 3.7b, uses a map as central part of the visualisation and utilises clustering to reduce graphical complexity. Their radial histogram glyphs are similar to the radial date histograms used in our approach in Sections 4.2.3 to 4.2.6.

We notice that from the radial visualisations, only one configuration of *Whisper* [10] and the visualisation by Drocourt et al. [15] use the angle to encode a direction. And even then, the solution used by Drocourt et al. is limited to visualise points that have a determinable order. We also realise that both visualisations use the distance from the centre to encode time, and not distance.

4 Concept

In this chapter we show the two core visualisation concepts we implemented. The first concept shown in Section 4.1 visualises geographical references within a single document. With the second concept, shown in Section 4.2, we explore possibilities to visualise the geographical and temporal distribution of documents in a corpus.

4.1 Visualising a Single Document with Geographical References

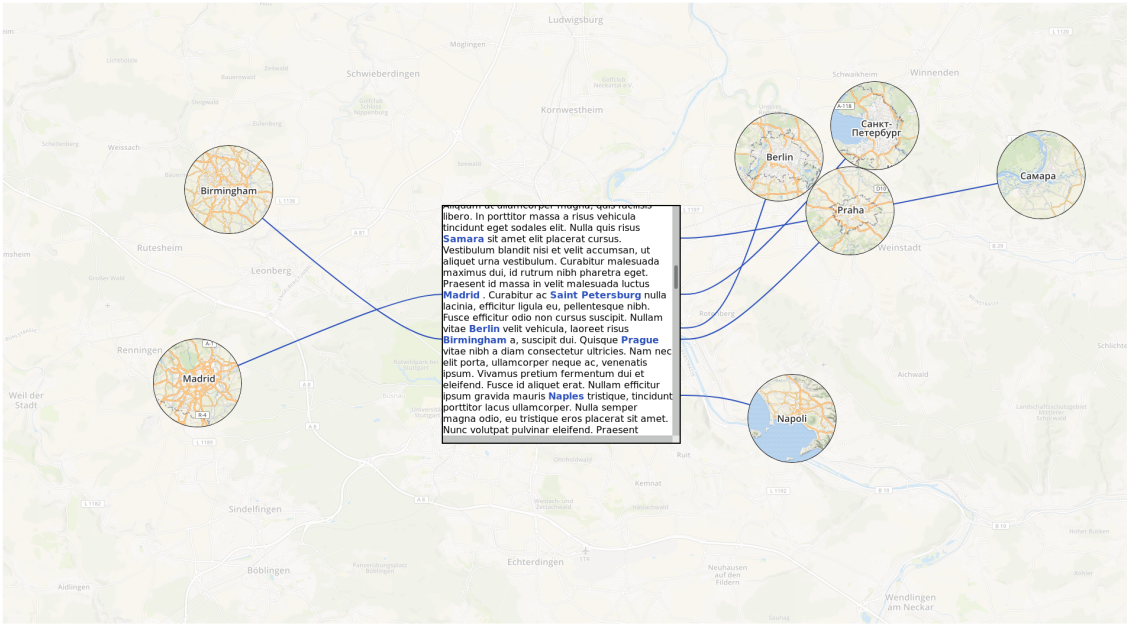


Figure 4.1: Implementation version *v0.4a*. A single text document is shown. In the background, a map of the text document's source location is visible. Geographical references in the text are highlighted, and map insets of the locations are shown and linked.

The first iteration of the prototype was aimed at visualising geographical references within one single document. To that end, we created a dummy document which contains city names at random places. The references are tagged with the cities' geographical coordinates.

Figure 4.1 shows a screenshot of the prototype in action. The prototype is implemented using JavaScript with D3.js, HTML and CSS. The text view in the middle is scrollable to allow larger texts to be shown. For each geo-location visible in the text view, a map inset is created showing a zoomed-in map of the location. The inset is connected to the text view via a cubic B-spline

such that the spline ends horizontally at the row of the text view where the location is mentioned. The background image shows the location where the text was published—in this case Stuttgart, Germany. The background map is shown with a opacity of only 30 % as to not distract from the core visualisation.

While scrolling through the text, insets appear and disappear with their corresponding mentions in the visible text passage. The splines connecting the text and the inset are updated when the row's vertical position changes.

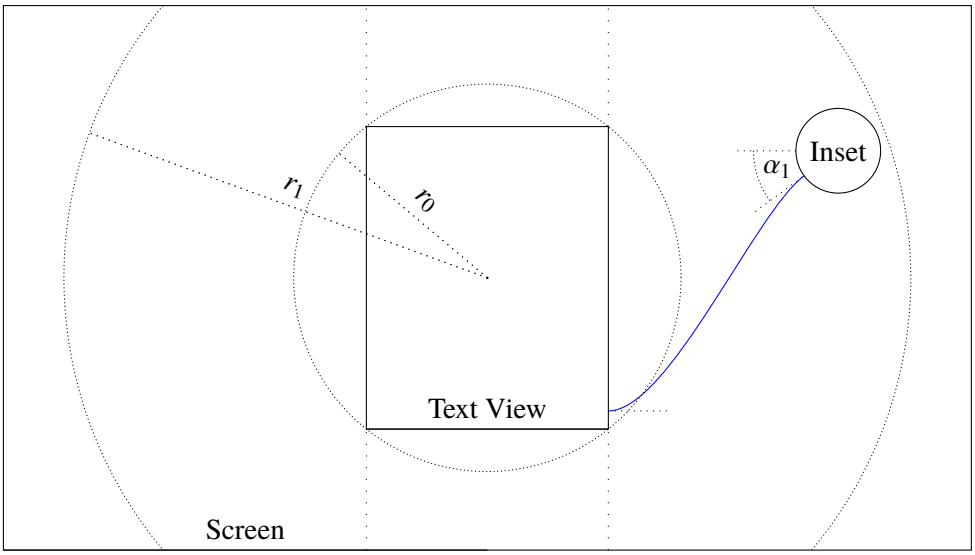


Figure 4.2: Measurements used in the placement of insets in *v0.4a*.

The insets are placed as follows: To reduce tension in the splines connecting the insets to the sides of the text view, the area above and below the text view is kept free on the width of the text view. In Figure 4.8 the evolution of this concept can be seen, visualising clearly how the central column is splitting up the circle around the centre location. A *minimum radius* r_0 is selected such that an inset placed at r_0 would no longer intersect the text view. The *maximum radius* r_1 is determined by the available screen real estate. For each location in the text, the great circle distance d and angle α_1 from the publication location is calculated. A *distance scale* s is calculated from the maximal distance $d_{\max}$ as follows:

$$s = \frac{d_{\max}}{r_1 - r_0}$$

Now, an inset i with great circle distance d and angle a_1 from the publication location is positioned at the position (x, y) where

$$x = r_0 + s \cdot d \cdot \cos \alpha_1$$
$$y = r_0 + s \cdot d \cdot \sin \alpha_1$$

An insets distance from the inner radius r_0 now is linearly proportionate to the distance from the document's publication location to the inset's location. In order to avoid insets overlapping, after the initial placement the insets are rearranged using a force directed layout with a small cutoff radius

while ensuring they are not pushed outside the visualisation area or into the text view. The spline's entry angle into the inset is $\alpha_1 + \pi$, such that the original direction the inset was placed in is still visible after final arrangement. In Figure 4.2, the measurements and angles are annotated.

This first version made it possible to visualise geographical references in a single text document. Also, a first sense of direction and distance of those locations can be conveyed to the user. However, as seen from the screenshot in Figure 4.1 as well as the diagram in Figure 4.2, the range in which the insets may be positioned is relatively small. Ideally, the size of r_0 would be nearly 0. Instead, it is nearly $\frac{r_1}{2}$. The linear mapping is thus distorted, especially paired with the size of the inset further reducing an exact reading.

Furthermore, as already argued in Section 2.2.3 and shown in Figure 2.5, the great circle direction might not be intuitive for the average user. The direction is further distorted by the dead areas above and below the text view. In case of overlaps between insets, the force layout step at the end further distorts the true direction. For the last point, the splines' end angles are supposed to be a partial remedy.

Using a well-known location as the origin from which the great circle angle and distance are calculated presumably makes it easier for the user to create a mental model of the locations mentioned in the text. We decided to use the document's publication location as origin. This way the location's position relative to that location would be meaningful. Another possibility would be to let the user chose a origin that is well-known to them, or use the current location of the user. This might be more intuitive to the user. However, for texts where the locations would not be spread out, but local, they might all have a very similar distance and direction from the chosen origin.

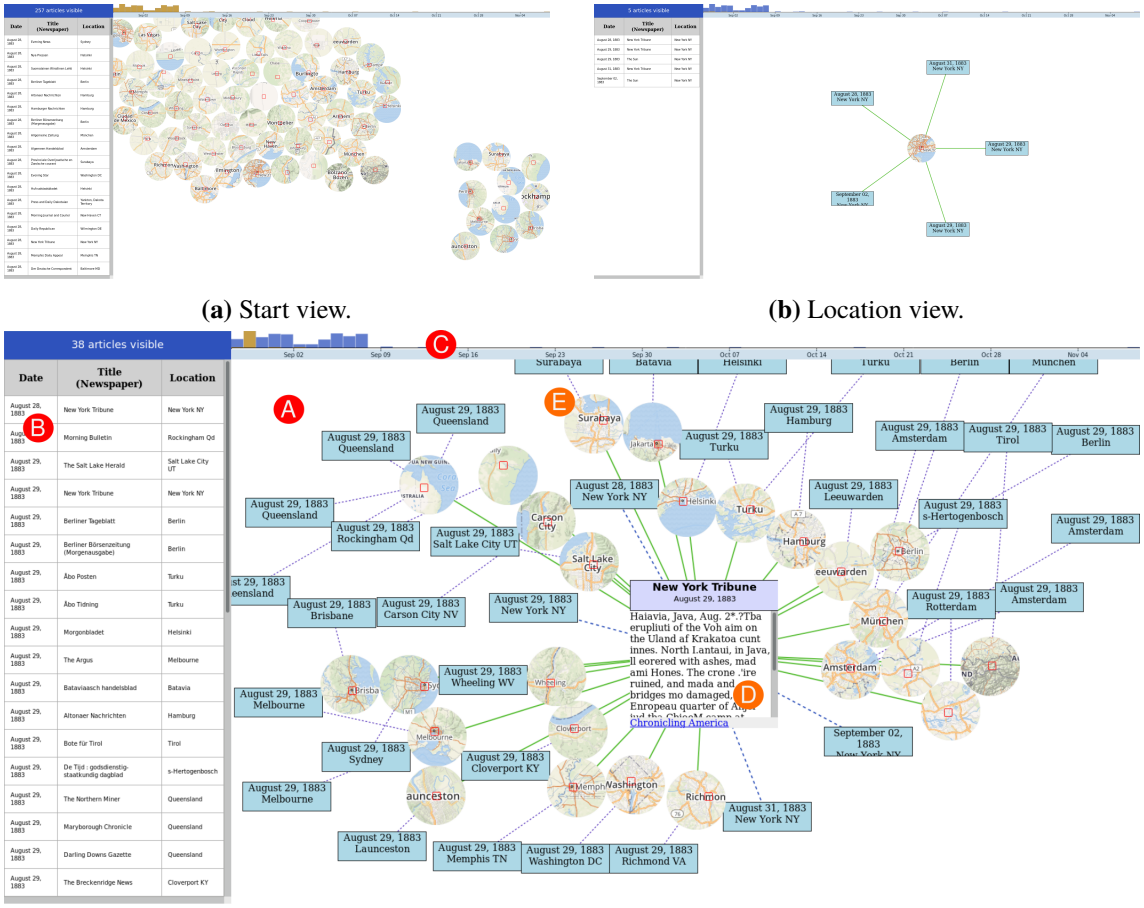
In conclusion, the first prototype has a few shortcomings which we address in the following versions. The core concept is kept, but later versions handle the distance and angle differently and are more robust with respect to the overdrawing problem. Additionally, they enable exploration of collections of text documents as opposed to the first version, which is limited to displaying one document at a time.

4.2 Visualising a Collection of Documents with Geographical References

In this section we use a dataset of 257 newspaper articles published during the autumn of 1883 in Europe, North America and Oceania. The dataset is described in further detail in Section 5.3.2. We analyse techniques for visualisation and exploration of datasets containing multiple documents with temporal and geographical references.

4.2.1 First Iteration: v0.6

The next prototype is used to visualise a collection of documents from different locations and dates. The visualisation is split up into three parts as seen in Figure 4.3: The main view, the list of documents and a histogram of publication dates. The main view itself changes for different stages of the visualisation, while the date histogram always shows the count of total and currently shown documents per day, and the document list shows a table of all currently visible documents.



(a) Start view.

(b) Location view.

(c) Neighbourhood graph view.

Figure 4.3: Implementation version *v0.6*. The visualisation consists of three parts. The central part (A) is augmented by a list of visible documents (B) on the left and a date histogram (C) on the top. The date histogram highlights the currently visible documents in yellow. In the *start view* (a) an inset is created for each location in which one or more documents were published. The insets are positioned again using great circle distance and direction and then arranged by force simulation to not overlap. Clicking on one of the insets switches to the *location view* (b) where the documents for one location are shown around it. Clicking on a document changes to the *neighbourhood graph view* (c). In this view, a document or location node is the central node of a tree graph (D). For a document, other documents from the same location or date are displayed, grouped by their location via an intermediate node. Clicking on a document node rebuilds the tree with that node in the centre, clicking a location node switches to the location view again, which is a special case of the neighbourhood graph view.

The starting point of the visualisation is visible in Figure 4.3a. In the main view, a map inset is created for each location in which a document was published. Those insets are then positioned using great circle distance and direction from the origin, which is selected to be the user's own location. To reduce overlapping of insets, the insets are then pushed around using repelling force simulation.

The main visualisation (A) is the *neighbourhood graph* shown in Figure 4.3c. It visualises a part of a graph

$$G(V, E)$$

where V is the set of all documents and for each two vertices v_1, v_2 there exists an edge

$$e = (v_1, v_2) \in E$$

if for a distance function dist

$$\text{dist} : V \times V \to \mathbb{R}_+$$

the distance of the two documents is lower than a threshold τ:

$$E = \left\{ (v_1, v_2) \,\middle|\, v_1 \in V,\ v_2 \in V,\ v_1 \neq v_2,\ \text{dist}(v_1, v_2) \leq \tau \right\}$$

For our prototype, we choose the distance function to return 0 if the two documents have the same publication location or publication date, and a number larger than the threshold otherwise. Implementing a more intricate distance function could be done in the future. The visualisation shows the currently selected document node v' large in the centre of the area, and all immediate neighbours $v \in N_{v'}$ of the node in the graph around it, where the neighbours are

$$N_{v'} = \left\{ v \in V \,\middle|\, (v, v') \in E \right\}$$

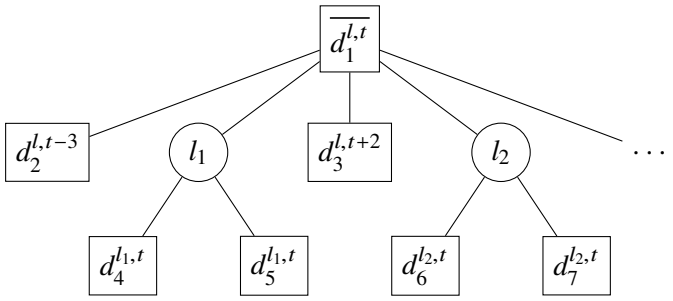


Figure 4.4: A segment of an example neighbourhood graph. The document d_1 was published at time t in the location l, and is shown in the centre. Around it, documents published in the same location (d_2, d_3) are placed. Documents published at the same time in different locations (d_4, d_5, d_6, d_7) are grouped by their locations (l_1, l_2).

The surrounding nodes of the same location are placed radially in close distance around the centre. For the nodes of other locations, a map inset of that location is created, via which the document nodes are connected to the centre node. A schema illustrating the hierarchy is shown in Figure 4.4. The connections have different line styles depending on whether they connect the centre to same-location

nodes, map inset nodes, or connect map inset nodes to documents of that location. The map inset nodes are placed based on the great circle distance and angle of their geographical location from the location of the centre node. In Figure 4.3c, the centre node's location is New York City, and the map insets from locations within the United States are arranged to the left (west) of it. Interestingly, the Surabaya, Indonesia map inset (E) is located nearly to the north, which is due to the fact that the great circle arc between New York City and Surabaya nearly passes the North Pole.

The timeline or date histogram (C) shows the number of documents per day. The currently visible documents are also highlighted in yellow. The data visible in Figure 4.3c mostly shows articles published on August 29th, 1883, and so that day's histogram bar is fully yellow. The date range visible can be changed by either clicking on a date or dragging the mouse over the timeline. Clicking a document will select that document as the centre node of the neighbourhood graph visualisation and reevaluate which other documents to show. This way, the user can traverse the graph, jumping to related documents. A sensible distance function should be chosen depending on the exploration task at hand.

Selecting a map inset will switch to the *location view* visible in Figure 4.3b. The selected map inset is the central node, and documents published in that location are placed radially around the inset. Selecting a document will return the visualisation to the neighbourhood graph view.

This version of the prototype can visualise a collection of documents. In the start view, the insets were no longer distorted into misleading locations by a central text view, as was the case in the first prototype. However, the sheer number of locations shown even for a small dataset of 257 documents meant the force-directed layout step moved nearly all insets around. In Figure 4.3a, this is apparent as the map inset clusters for North America and Europe have merged, and for instance Mexico City is placed to the northwest of Baltimore. Here, a better way to deal with overlapping insets is required, which we attempt in the later versions discussed in Sections 4.2.2 to 4.2.6.

The neighbourhood graph view introduced a way to navigate related documents and explore a collection. However, it also suffers from crowding as described by Rosenholtz et al [32]. Here, the overlapping issue also needed to be tackled differently, which we also address in later versions. The core concept of traversing a graph node by node has proven useful. The selection of a sensible distance function should be explored further. For instance, incorporating great circle distance in the calculation of document distance would show documents published in close-by locations as well. This would be interesting as geographical closeness could indicate similarity as by Tobler's first law of geography: '*Everything is related to everything else, but near things are more related than distant things*' [40, p. 236]. Similarity in date, topics discussed or names and geonames mentioned could also be considered.

4.2.2 Introducing a Word Cloud: v0.7

For the next iteration of the visualisation, the start view (Figure 4.3a) is discarded. Instead, a word cloud (Figure 4.5a) is shown. For this, the documents of the multilingual sample dataset are all automatically translated into English. Then, after removal of stop words, the term frequency of all words is determined, and the top 200 words[1] are drawn in the word cloud. First use of brushing

[1] The word cloud placement algorithm will attempt to place 200 words in the word cloud. It will however terminate before if no more free space can be found for further words.

(a) Word cloud view. **(b)** Single word view.

Figure 4.5: Implementation version *v0.7*. In the *word cloud view* (a) a word cloud of the most used words in all documents (sans stop words) is shown. On hovering over a word, it and the data it represents is brushed in the word cloud and the date histogram. Clicking on a word switches to the *single word view* (b), where the word is shown in the centre and the document nodes containing the word are drawn around it.

and linking is implemented by highlighting parts of the date histogram when hovering over a word in the word cloud. Clicking on a word switches to a single word view (Figure 4.5b). Around the selected word, the nodes of all documents containing the word are drawn. Interaction with the document nodes then leads to the neighbourhood graph introduced in Section 4.2.1.

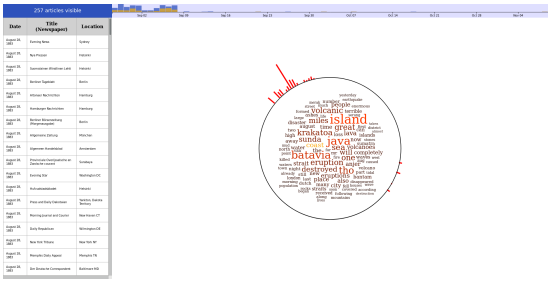
The word cloud as overview no longer holds any geographical information, but gives a better overview concerning the topics of the document collection. Combined with the brushing and linking interaction, it is now possible to interactively explore the temporal distribution of single words. Ideally, in future versions also the geographical distributions of word usage are visualised. This is realised in later versions described in Sections 4.2.4 to 4.2.6. Another solution would be a tag map as is used by Jaffe et al. [25], in *WorldExplorer* [1] or *ScatterBlogs* [6].

The automatic translation was done by using the Google Translate API [48], letting it automatically determine the language of the document, and translating it into English. Automatic translation is not perfect and often struggles, especially with historical texts and texts containing OCR scanning errors. For the creation of a word cloud, automatic translation was however sufficient, especially for a prototype. Ideally, a second step of preprocessing would be to change all words into their word stem, but even without that step the process yields satisfactory results.

The single word view turns out to become very crowded even for a small number of articles, as already seen in Figure 4.5b. This view calls for some aggregation to happen before visualising, which is done in later versions. Using term frequency for the word cloud also merits second thought, as words that are used in nearly all documents are probably not as interesting as words that are used often in a small set of documents. To that end, using *tf-idf* instead of term frequency would yield better results. For certain tasks, using the document frequency might also work well.

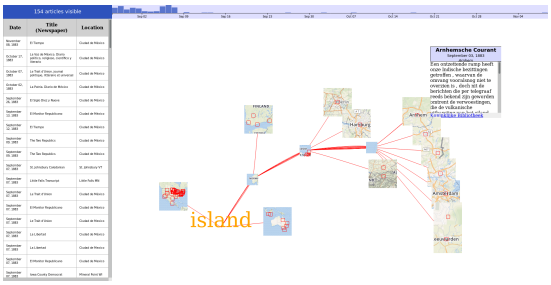
4.2.3 Introducing Hierarchical Clustering: v0.8b

In the next version of the prototype, we shrink the word cloud to a smaller area in the centre of the visualisation and added a radial histogram around it. For a brushed word in the word cloud, the great circle direction from a chosen position—the geographical location of the user—to each containing document's publication location is calculated. Binned into 180 ranges each spanning 2°,

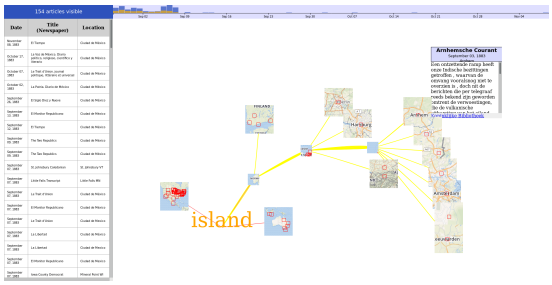


(a) The start view is a word cloud with a radial histogram.

(b) Three top-level location clusters are shown for a word.



(c) A subtree is expanded out into a document node.



(d) A subtree is brushed.

Figure 4.6: Implementation version *v0.8b*. The word cloud starting point is now encased in a radial histogram (a) which shows a histogram of the great circle angles from an origin to the documents containing the brushed word. The neighbourhood graph view is replaced by a tree visualisation based on hierarchical clustering of the locations of documents containing a selected word (b–d). Whole subtrees of the visualisation can be brushed simultaneously (d).

the count of documents per range is encoded into a bar of the radial histogram. To match the scaling of the histogram bars, the word cloud is now constructed using the document frequency. The user can now interactively explore which words are used in documents in which direction, an example of which is shown in Figure 4.6a.

For the second view seen in Figures 4.6b to 4.6d, the publication locations are first hierarchically clustered using the great circle distance as distance function. The resulting dendrogram is then cut off at a high threshold, leading to a number of subtree dendrograms. This step is repeated a few times with the resulting subtrees, such that sensible hierarchical clusters are created. The thresholds used for this dataset are 5000 km, 1000 km and 200 km. The first threshold of 5000 km results in three clusters as apparent from Figure 4.6b: North America, Europe and Oceania.

The resulting tree is visualised by initially showing the top-level subtrees in a folded state. By clicking on a subtree, it expands one level, showing its own subtrees. To prevent neighbouring subtrees from overlapping visually, only one subtree is expanded at a time. Only the top-level siblings of each expanded subtree are visible, as shown in Figures 4.6c to 4.6d. This can be considered a variant of the generalised fisheye view discussed by Furnas [18]. The top level cluster nodes are placed in a constant distance from the centre, using the great circle angle from a location chosen by the user to the centre of the Cartesian bounding box of the locations in the cluster. The lower level cluster nodes are placed in arcs outwards from their parent node, regardless of their

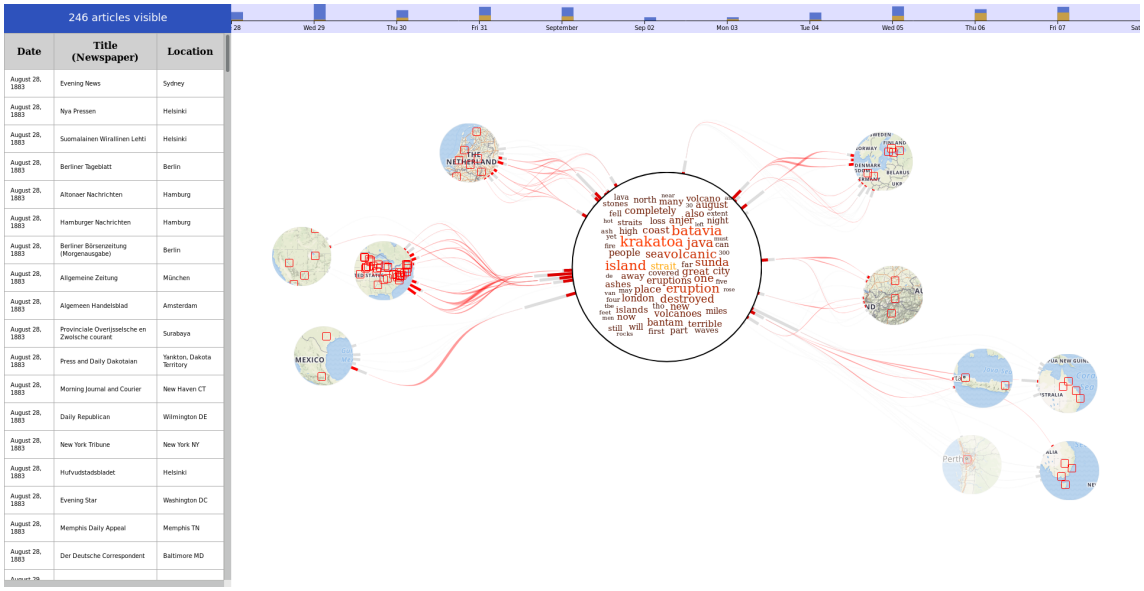


Figure 4.7: Implementation version *v0.9b*. The map insets are placed around the word cloud based on their direction and distance from the user's location. A radial date histogram is drawn around each inset, and the histogram bars are connected using B-splines.

geographical location, as that location is within the bounding box of the parent node in any case. The cluster nodes as well as the document nodes again support brushing and linking, highlighting parts of the date histogram. Brushing a subtree node highlights all documents of that subtree.

Having a direction-indicating radial histogram around the word cloud enables the user to explore the usage of words in different directions from his own point of view. However, that view does not show the real locations yet, which was amended in the prototype version introduced in Section 4.2.4. The hierarchical clustering of the locations fixes the problems with overlapping and distortion from the force-directed layout step used in the previous versions. Additionally, it is now possible to get an overview of regions and then drill down into smaller regions, single locations and finally documents. Through the interactions and linking and brushing of the tree view, exploration of geographical distributions is facilitated.

4.2.4 Combining the Views: v0.9b

The next version, visible in Figure 4.7, adds map insets around the word cloud. Those are placed using equirectangular direction and great circle distance from the origin to the inset's bounding box' centre, leaving the centre free as shown in Figure 4.2. Around the map insets, a radial date histogram is drawn based on the documents published in the respective locations. The radial histograms span 90°, are arranged clockwise and face the centre of the visualisation.

Besides the linking of the date histogram, the radial histograms and the word cloud, this version of the prototype also connects parts of the visualisation. The bars of the directional radial histogram around the word cloud are connected by B-splines to the radial date histograms' bars of the map insets. As locations of one inset can have a slightly different direction from the origin, and

documents from one location can have different publishing dates, this is a many-to-many mapping. The B-splines are also linked, changing their thickness in the middle and being drawn in a different colour. The thickness correlates with the number of documents linked with that date and direction.

The hierarchical tree introduced in the previous version in Section 4.2.3 is no longer present in this version. Later versions introduced in Section 4.2.5 use a different method of drilling down in the hierarchy.

The brushing and linking in this version of the prototype facilitate exploring the temporal as well as the spatial distribution of documents and topics. This can be done interactively by hovering the mouse over different parts of the visualisation.

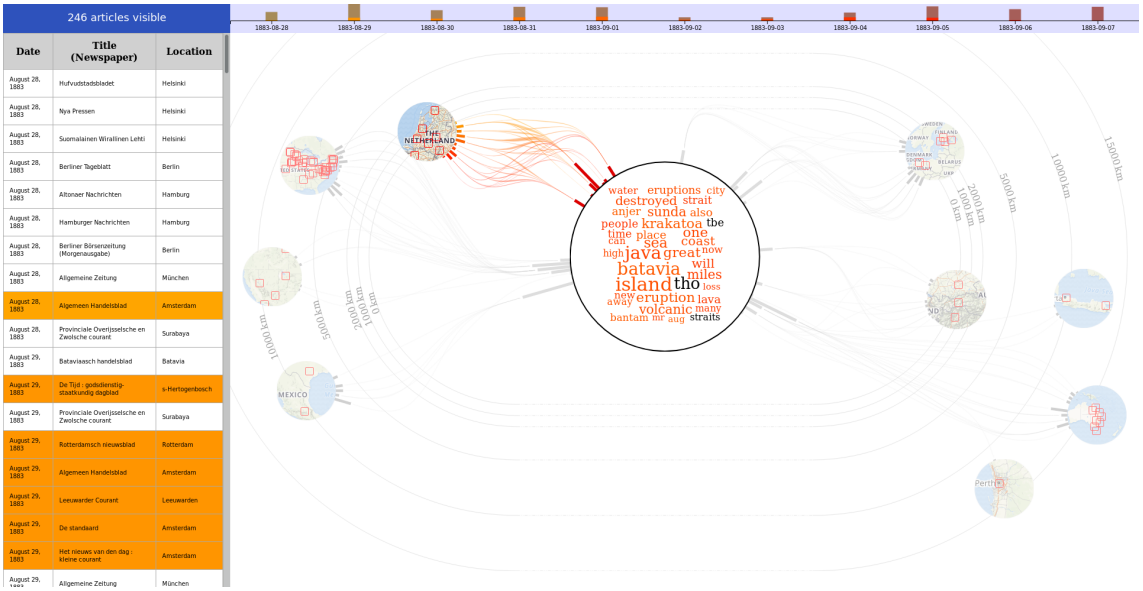
4.2.5 Adding Visual Cues: v0.12a

The next versions introduce a colour scale for the date. The first date is encoded as orange, and the last as red. All date-specific linking is done in the respective colour. An example, showing the colouring of the splines and radial date histograms, can be seen in Figure 4.9. This aims to give the user a better distinction on the temporal distribution of the data.

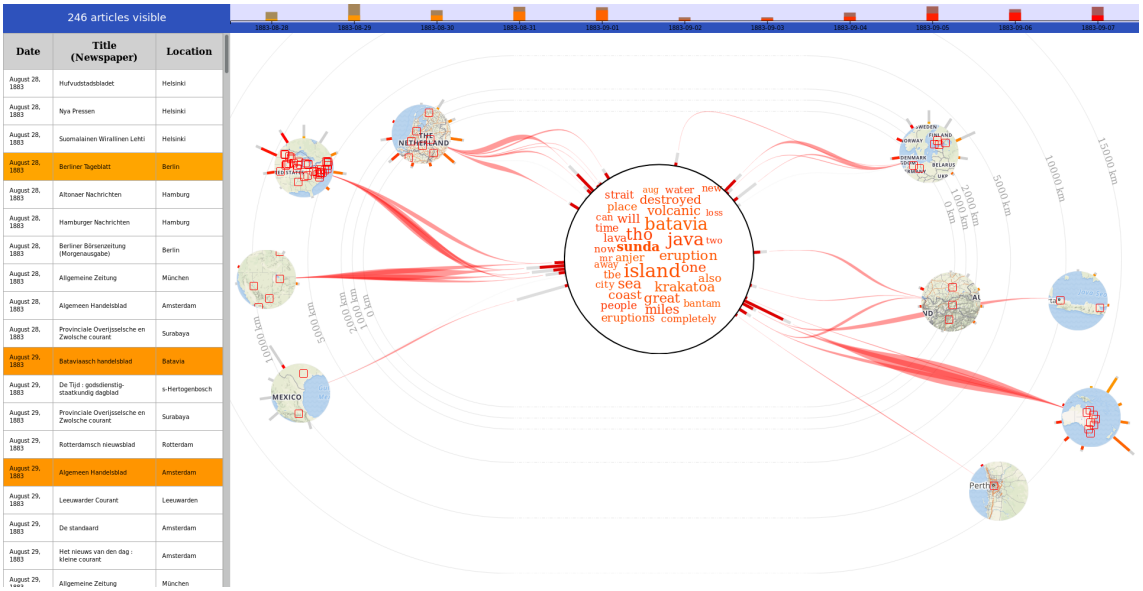
Another new feature is the addition of distance isolines, visible in Figure 4.8. The goal of those is to give the users a better indication of the distance from their position to the locations in the map insets. This is necessary because of the large offsets in the linear scaling. In addition to drawing the isolines, the force-directed layout step at the end of the inset placement makes sure that the insets stay on the isoline representing their distance from the origin. Now, only the direction can be distorted by the last layout step, and no longer the distance.

Additionally, everything in the visualisation can now be hovered over or clicked. On hovering, the represented data is linked throughout the whole visualisation. On clicking, the visualisation is restarted using only the data aggregated in the selected element. This creates a new way to drill down into the data during exploration. The drill down steps can easily be reversed by using the *back* button left of the document counter, following the sixth of Shneiderman's guidelines when designing a user interface: '*Permit easy reversal of actions*' [34, p. 75]. The transition from Figure 4.10a to Figure 4.10b shows such a drill down in action, reducing the visible data from 246 documents to 145. In Figure 4.10b, the back button is visible.

We try out a variant of the radial date histograms as well. We provide a flag to be set so that the radial histograms would span all 360° around the map insets without facing towards the centre, but instead always start at 12 o'clock. Subsequently, there is only one B-spline from each relevant bar of the directional radial histogram to an inset. This also means the splines no longer are coloured according to the day they represent, as they no longer represent single days. This version is shown in Figure 4.8b. We conclude that while having all radial date histograms start in the same direction was useful, having dedicated links for each day is more so. As a result, the many-to-many variant is continued in the following version. However, the flag—and thus the possibility to use the second variant—is kept.



(a) Many-to-many linking between the radial date histograms and the radial directional histogram. The radial date histograms are facing the centre of the visualisation and the word cloud.



(b) Many-to-one linking between the radial directional histogram in the centre and the map insets. The radial date histograms start at 12 o'clock and continue clockwise.

Figure 4.8: Implementation version *v0.12a*. Colour is used to indicate time, starting in orange and ending in red. The radial directional histogram around the word cloud is connected many-to-many (a) or many-to-one (b) to the map insets and their radial date histograms.

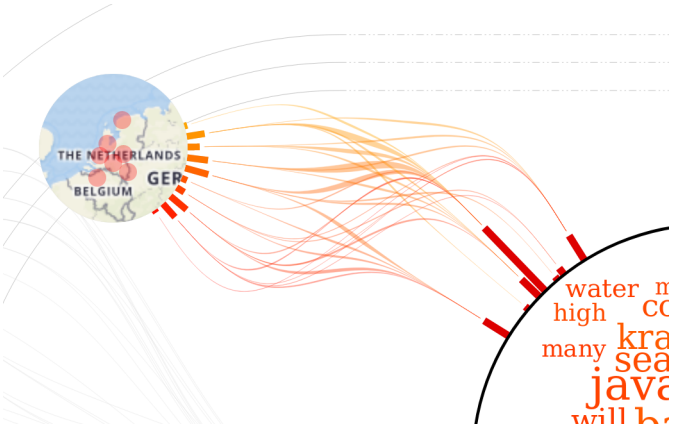


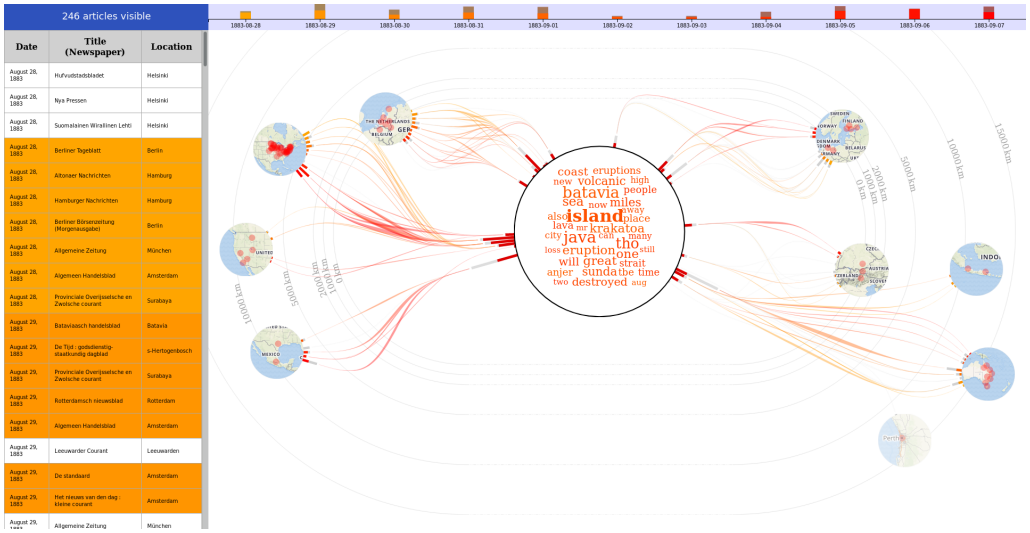
Figure 4.9: The splines and radial date histograms are colour-coded based on the date they represent. As the directional histogram's bars around the word cloud do not represent singular dates, their colour is uninfluenced.

4.2.6 Adding Single Document View: v1.0

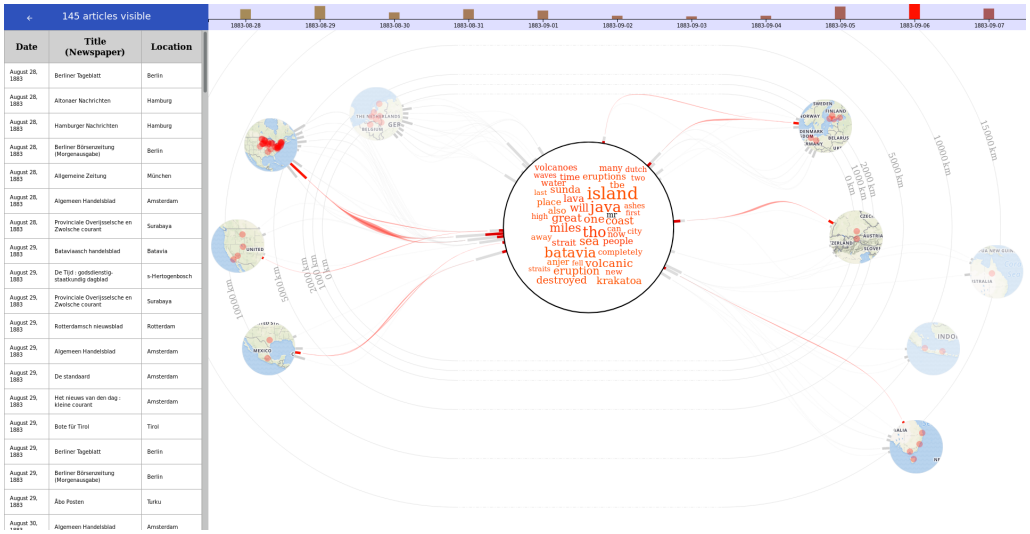
The final version of the prototype does not introduce many changes. For one, the markers in the insets are changed and made opaque, so that dense areas in the map, such as the east coast of the United States (Figures 4.10a and 4.10b) can be identified as such. And second, clicking on a document opens that document in a second window, shown in Figure 4.10c. This view shows the document's publication date, location, and source in the title, followed by the text itself and a map showing the publication location.

Opening single documents in a different window, independent of the main interface, means that the user can keep documents open while continuing exploration of the dataset. Management of open documents is thus left to the user and his operating system.

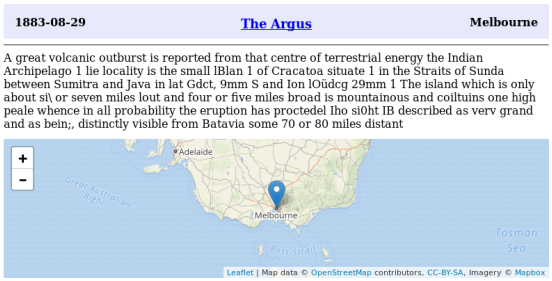
Future work could include incorporating the first prototype shown in Section 4.1 on page 29 in the single document view. That would however require to first find and tag geographical references in the document texts.



(a) Initial view. A word in the word cloud is brushed and linked.



(b) The dataset was drilled down into the word '*island*'. A day in the date histogram is brushed and linked.



(c) The single document window, showing the document and its publication location.

Figure 4.10: Implementation version *v1.0*. In (a) and (b), the main view is shown together with the drill down mechanic. In (b), the back button can be seen in the top left, which goes one step back in the drill down stack. In (c), the single document view is shown for a document published in Melbourne, Australia, together with an overview map and publication details.

5 Implementation

In this chapter, we describe the libraries and tools used to create the prototypes. We also explain the implementation of the back end logic of the server providing the web page. Furthermore, the datasets that were used are described, and the preprocessing steps are listed.

5.1 Front End Implementation

In the following, we will describe the languages and libraries used to implement the prototypes' front ends.

5.1.1 JavaScript and TypeScript

The versions introduced in Section 4.1 and in Sections 4.2.1 to 4.2.3 are implemented using HTML, CSS and pure client-side JavaScript (ECMAScript). For subsequent versions, the existing code has been ported to TypeScript. This has helped cleaning up the codebase, modularising it and separating concerns between classes and modules. It also requires there to be a transpilation step, which translates and bundles all TypeScript modules and all external JavaScript library dependencies into a single JavaScript file, which is then loaded by the browser.

TypeScript is an extension of JavaScript which adds better support for object-oriented programming and type safety. For example, variables and function parameters are annotated with their type. Type bindings are provided for most popular JavaScript libraries, so that those can be used for TypeScript development without greater circumstances. The transpilation step also issues warnings and errors and reveals bugs which would have passed unnoticed using pure JavaScript, which is strictly a duck-typing language.

5.1.2 D3.js

D3.js [44], short for *Data Driven Documents*, is a JavaScript library that allows the programmer to bind data to the Document Object Model (DOM).

The document can then be manipulated easily based on the data. D3.js allows quick and feature-rich manipulation of web pages using HTML, CSS, and *Scalable Vector Graphics* (SVG). D3.js provides the backbone of our visualisation's implementation, starting with the loading of the data and continuing with the creation of the HTML and SVG representation.

```
1  // Select all table cells in tables of class 'financial' with negative value
2  var selection = d3.selectAll('table.financial tr td')
3                    .filter(function() {
4                      let cell = d3.select(this);     // 'this' overridden by function
5                      let text = cell.text();
6                      try {
7                        let val = parseFloat(text);
8                        return (val < 0);
9                      } catch (err) {
10                       // cell does not contain a number
11                       return false;
12                     }
13                   });
14
15 // Give those cells a red background
16 selection.style('background', 'red');
```

Listing 5.1: Selections and dynamic styling using D3.js.

A core feature of D3.js is *selections*. A selection can be a single DOM element or a set of elements matching the specified criteria. A rather involved example for a selection is shown in Listing 5.1. This selection is saved into a variable, which can then be used later. The selection starts by selecting all table cells (`td`) contained in table rows (`tr`) contained in tables of class `financial` (`table.financial`). The selection is then filtered using an in-place function, which is called for all nodes of the selection and evaluates if their text is a negative number.

Listing 5.1 also shows how D3.js can apply styles and transformations on whole selections. In the listing, the selection's nodes get a red background. It is also possible to pass a function to the `.style()` or `.attr()` calls, such that individual values can be computed for each node based on its properties.

Changing the data on a selection is also a powerful feature of D3.js. When changing the data, three subselections are created: Elements of the original selection whose data is updated, elements that are new (`.enter()`) and elements that are removed from the data (`.exit()`). The `.enter()` and `.exit()` calls on selections, combined with unique identification of a datum, makes D3.js a potent tool when dynamically displaying data. Attribute and style modifications on selections can even be wrapped in transitions. D3.js will then interpolate between old and new values and provide a gradual transition. This is possible for number value changes as well as colour changes, and is helpful for the user because there is no loss of context when the displayed elements change.

5.1.3 Other JavaScript Libraries

For the creation of map elements, *Leaflet.js* [42] is used. Leaflet provides a simple API to place a map into a DOM element. The style and controls of the map element can be modified in many ways, including creating markers in the map and fitting the map to the bounding box of a set of geolocations. For retrieval of the actual map tiles, *MapBox* [50] is used.

The early versions in Section 4.2 use *Crossfilter* [53]. Crossfilter provides utilities for multidimensional filtering in a dataset, which is useful for drill down operations on a dataset. For the drill down operations introduced in Section 4.2.5, the Crossfilter API proved to be too restrictive in the amount of combined filters allowed, and Crossfilter was almost completely replaced by a custom solution in the approach.

The hierarchical clustering shown in Sections 4.2.3 to 4.2.6 is implemented using a JavaScript library for hierarchical clustering named *clusterfck* [43]. Clusterfck provides a single method `hcluster`, which takes a list of elements, a distance function, a linkage criterion and an optional threshold, and returns a binary tree, the dendrogram. This tree then has to be flattened to result in the graphs displayed in Sections 4.2.3 to 4.2.6.

For the creation of the word cloud, *d3-cloud* [46] is used. d3-cloud is not strictly part of D3.js, but has a very similar interface. It allows the simple creation of word clouds. However, it is only possible to create word clouds with rectangular bounds, which proves to be a problem for the prototypes introduced in Sections 4.2.3 to 4.2.6, as can be seen from the word 'new' intersecting the circle in Figure 4.8b. Using a word cloud placement strategy that respects circular bounds would be a possible improvement for the prototype.

5.2 The Server Back End and REST API

The server back end of the web page has gradually evolved from a static file server to a REST API. In the first prototypes, the JavaScript, HTML and CSS content are served as static pages, and the external libraries are loaded from their respective online locations. For the prototypes introduced in Section 4.2, the geocoding information is first queried directly from the Google Geocoding API [49] for each datum and each reload of the web page. However, the Google Geocoding API only allowed for 5,000 requests per day at the time[1], which we exhausted at times during development and testing. To reduce the usage of the Google Geocoding API, the geographical information for the locations in the dataset has been downloaded and stored offline, and the dataset annotated with the unique IDs of the geolocations as described in Section 5.3.3. The cached geolocation data is also statically served by the back end.

The server back end's first realisation use the Linux `php` package, and later `devd`, which both create a static HTTP server serving the current directory's files. This is sufficient for the versions that only serve static files. Later versions of the prototype use utility REST API calls, which require a server that can handle those. Especially the last version introduced in Section 4.2.6 requires some additional logic in the server back end. Specifically, it was required to be able to view a single document in a new window, which means opening a window with a URL pointing to a REST endpoint on the server.

In order to realise this, we implement the server back end using *Flask* [51], a Python library that greatly facilitates the implementation of a web server serving both static and dynamic content. Flask uses the *Jinja2* [52] template engine, which means the single document web page can be templated so the server only needs to fill in the texts at the appropriate locations.

[1] As of June 11th, 2018, the old pricing model for the Google Geocoding API was retired. Now, a user has one free API request per day and pays 0.005 USD for each subsequent request.

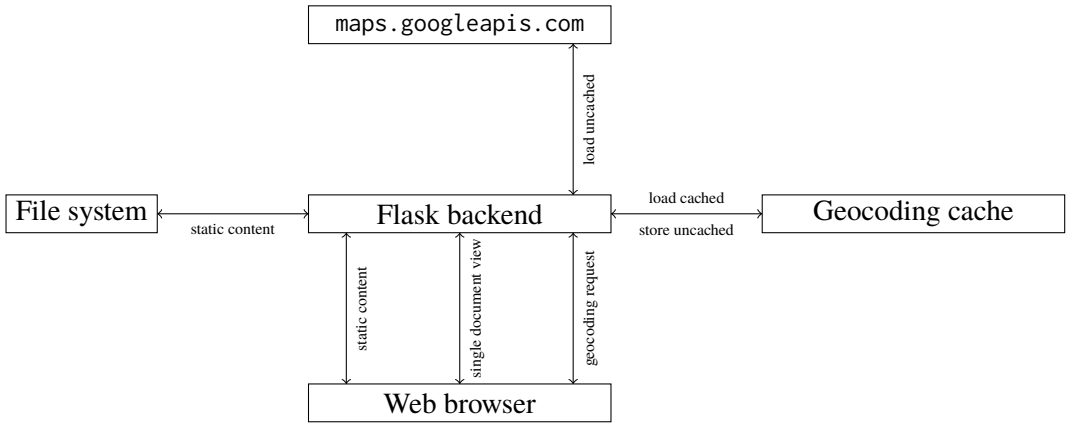


Figure 5.1: The client-server architecture used for later versions of the prototype. Static content is served directly from the file system. Dynamic content, such as geocoding information and single document pages, are generated by the server. The geocoding cache contains the geolocation information for all documents contained in the dataset described in Section 5.3.2. For future use with other datasets, the server must be augmented to load uncached geocoding information from the Google Geocoding API or another comparable service and cache it for subsequent requests.

Some intermediary versions of the prototype also mirror the Google Geocoding API, so that the API call can just be redirected to the own server. For the later versions, the cached geolocations are simply served as a static list of JSON objects. For future versions, where the locations are not known beforehand, this could be reimplemented as seen in Figure 5.1: Geocoding requests are made to the own server, which provides a cached answer if it exists, and forwards the request to the Google Geocoding API otherwise and caches and forwards the reply. Especially with the new pricing model of the API, not resending all requests each time the user reloads the page seems prudent.

5.3 Used Datasets

Different datasets are used for the single prototype described in Section 4.1 and for the subsequent prototypes depicted in Section 4.2. In this section we describe the two datasets and the preprocessing steps taken. We also list the difficulties met when handling real-world data.

5.3.1 Dataset for Visualising a Single Document

```
1 Vivamus elit lacus, elementum in sem a, ultrices luctus <geotag id="odessa" lat="46.466667"
2 lon="30.733333">Odessa</geotag> turpis. Vestibulum ante ipsum primis in faucibus orci luctus et
3 ultrices posuere cubilia Curae; Nullam scelerisque enim felis, ac bibendum ligula tempus non. Cras
4 convallis erat eros <geotag id="moscow" lat="55.75" lon="37.616667">Moscow</geotag>, in
5 pellentesque arcu iaculis at. Curabitur maximus sapien metus, et
```

Listing 5.2: Extract of the data set used in *v0.4a*.

The dataset used in the implementation introduced in Section 4.1 is a hand-crafted test dataset. The text is a part of the ubiquitous *Lorem Ipsum* text, which is based on a Latin text by Cicero. Scattered through that text are the names of large cities across the European continent, which are annotated with their geographical coordinates using HTML-tags as shown in Listing 5.2. Those tags themselves are not visible when rendering the text in a web browser, only their content—the city name—is shown. The rest of the data is still part of the HTML source, and can be manipulated and extracted using D3.js.

5.3.2 Dataset for Visualising a Collection of Documents

The dataset used during implementation of the prototypes introduced in Section 4.2 is a collection of 257 digitised historic newspaper articles. Those articles were published by newspapers in North America, Europe and Oceania between August 28th and November 8th, 1883, with most of the articles published before November 9th.

The topic of the articles is the eruption of the volcanic island *Krakatoa* in the Sunda Street in today's Indonesia. That eruption was heard thousands of kilometres away, killed over 36,000 people, and the sound wave was measured rounding the earth three and a half times, based on Symons et al. [39, p. 63]. Additionally, the ashes in the atmosphere lowered the average global temperature and the temperature and weather patterns showed irregularities until 1888 according to data provided by Bradley [7].

Table 5.1: Data fields of the dataset used during implementation of the prototypes visualising a collection of documents.

Field	Description
Date	Date of publication.
Title (Newspaper)	Title of the newspaper.
Location	Location of publication.
Search term	Search term used to find the article.
Text	Original text[2] of the article.
Language	Original language of the article's text.
Corpus	Corpus containing the document.
Link	A URL to the scanned document.

The dataset is extracted from data collected by the *Oceanic Exchanges* project and provided in the context of a cooperation with the Institute of Literary Studies at the University of Stuttgart. It contains the data fields listed in Table 5.1. Before use, the data has been cleaned and supplemented as described in Section 5.3.3. In the last years, text recognition by computers (OCR – Optical Character Recognition) has gotten good enough that large amounts of old newspapers and documents could be digitised. This has also increased the demand for tools that can be used to sift and explore those mountains of data and bring the documents in relation to each other. Our prototype aims to help in that regard.

[2]The texts are obtained by scanning old newspapers, and often contain OCR (optical character recognition) errors, which results in text where single characters or words are not matching the context.

However, there are still problems to tackle. One of these problems is that OCR still makes errors, which result in irregularities in the scanned texts. Examining the newspaper article displayed in Figure 4.10c reveals quite a few such errors. The source of such errors could be the typeface used when printing, dirt on the document, a partially destroyed document or a blurred scanning image, to just name a few problems. Automating the finding and amending of such errors is not trivial, and in the end human interaction is probably necessary to give a scanned document a last polish.

5.3.3 Preprocessing of the Data

Table 5.2: Additional data fields (to Table 5.1) after preprocessing.

Field	Description
Index	Incremental index of the article.
place_id	Unique geolocation ID used by the Google Geocoding API.
translated	Rudimentary translation of text into English.

Some irregularities in the dataset, as well as some missing information, required some preprocessing. To start, the data was assembled by multiple people in multiple countries, and so the dates were in different locales and different formatting, which needed to be normalised into the `YYYY-MM-DD` format.

Some additions to the data are also required. Those are listed in Table 5.2. First, the publication locations have to be processed. Using the Google Geocoding API [49], the location names have been looked up. The Google Geocoding API among other things returns a unique ID for each geolocation, which has been appended to the data. Several datums' locations were entered wrong and have been corrected. The geocoding results for the locations in the dataset have then been stored locally, so they can be queried for each datum using the `place_id`.

For the versions including a word cloud in Sections 4.2.2 to 4.2.6, all texts needed to be in the same language. Therefore, a data field `translated` has been added. Using the Google Translation API [48], the non-English texts are translated into English and the translated texts appended. Because of OCR errors and a generally dated language, the translated texts are not perfect, but good enough for the construction of a word cloud.

For the word cloud, the most common OCR errors are also added to the list of stop words. This includes multiple punctuation characters as well as symbols used to separate paragraphs in the newspaper. As can be seen from words like '*tbe*' or '*tho*', OCR errors of the stop word '*the*', which can be found for instance in Figure 4.10b, not all OCR errors have been removed. This also indicates that there is still need for human interaction and that data cleaning processes are merely computer-aided and not strictly automated.

6 Evaluation

In this chapter, we present some usages examples for the prototype and show screenshots. We also conduct an expert study with a single stakeholder, and draw conclusions on what results were reached with this work.

6.1 Usage Examples

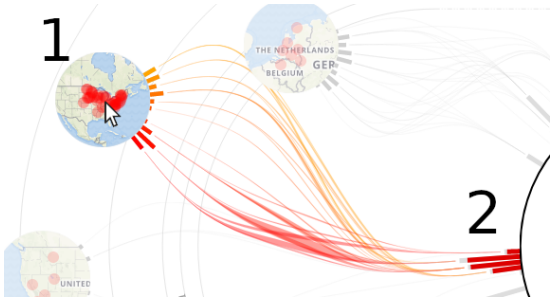
We present three simple usage examples for the prototype. All three aim to solve a task for the Krakatoa dataset. we describe the steps we take to solve the tasks, and provide screenshots of the visualisation showing the progress.

6.1.1 Comparing Spatial Distribution of Word Usage

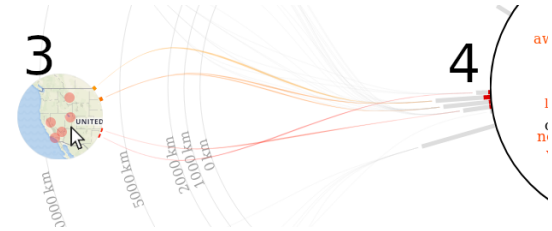
In the first usage example, we want to find out how the use of the word '*coast*' differs between articles published in the United States of America and those published in Australia. From the initial view, we locate the first map inset that contains locations in the USA (1) in Figure 6.1a. We hover the mouse over that inset and then locate the directional histogram bars that get linked (2). In Figure 6.1b, we then locate the second map inset (3) which contains locations in the USA, and hover the mouse over that inset. We find that the same directional histogram bars are linked (4). Following the greyed-out splines, we also conclude that these histogram bars only connect to locations in the USA. For the map insets representing Australia (5, 7) in Figures 6.1c and 6.1d, we do the same. We conclude that the directional histogram bars (6, 8) are only connected to locations in Australia.

We then locate the word '*coast*' (9) in the word cloud in Figure 6.1e. We hover over it with the mouse and inspect the directional histogram bars that we found being connected to those locations (10, 11). By comparing the percentages of the bars filled red—the portion of articles represented by the bar that are currently being linked—we can conclude that the word '*coast*' is used more often by newspapers in the USA than by newspapers in Australia.

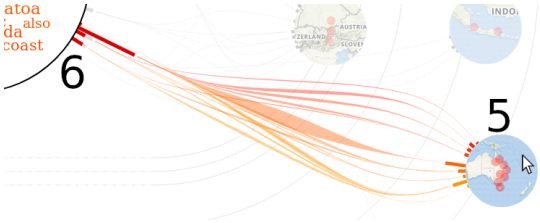
When doing this for map insets sharing directional histogram bars with other map insets that are not part of the query, we would have to directly compare the histogram bars on the map insets instead. Alternatively, we could compare the thickness of the splines connecting to the map insets. Solving this task for locations that share their map insets with other locations is not yet possible with this method. In Section 8.2, we discuss future improvements that would allow arbitrary selections, as well as the possibility to let the hierarchical clustering respect country borders.



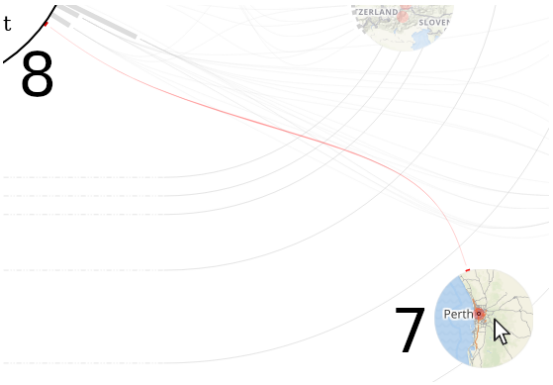
(a) Eastern and central USA inset selected.



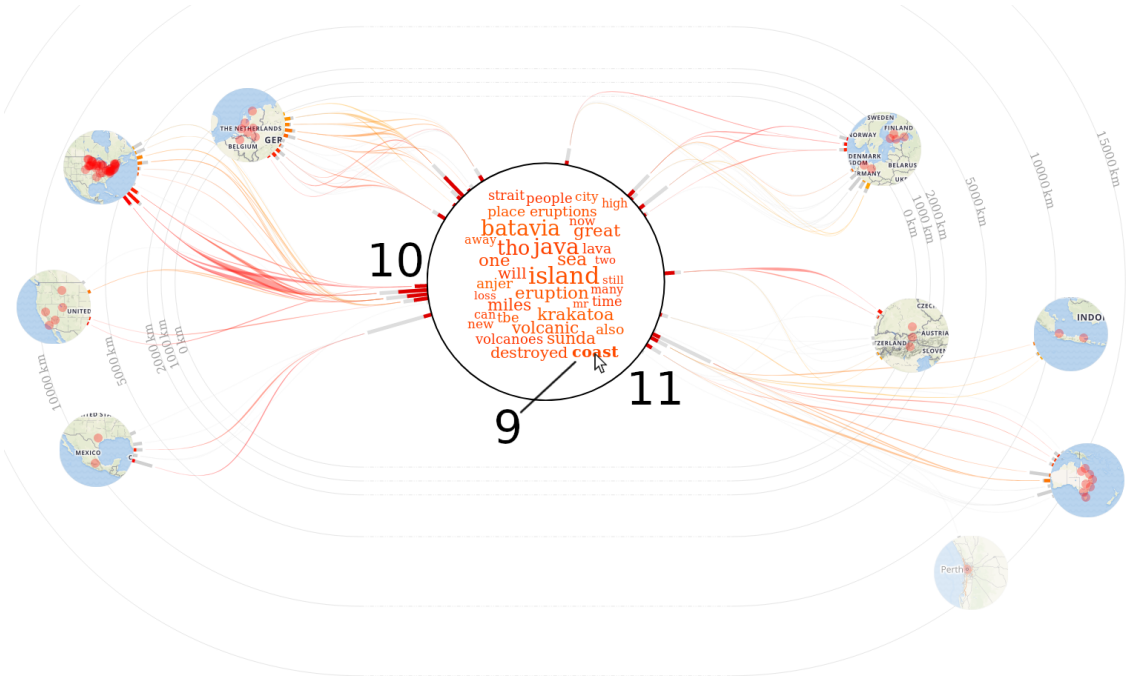
(b) Western USA inset selected.



(c) Eastern Australia inset selected.

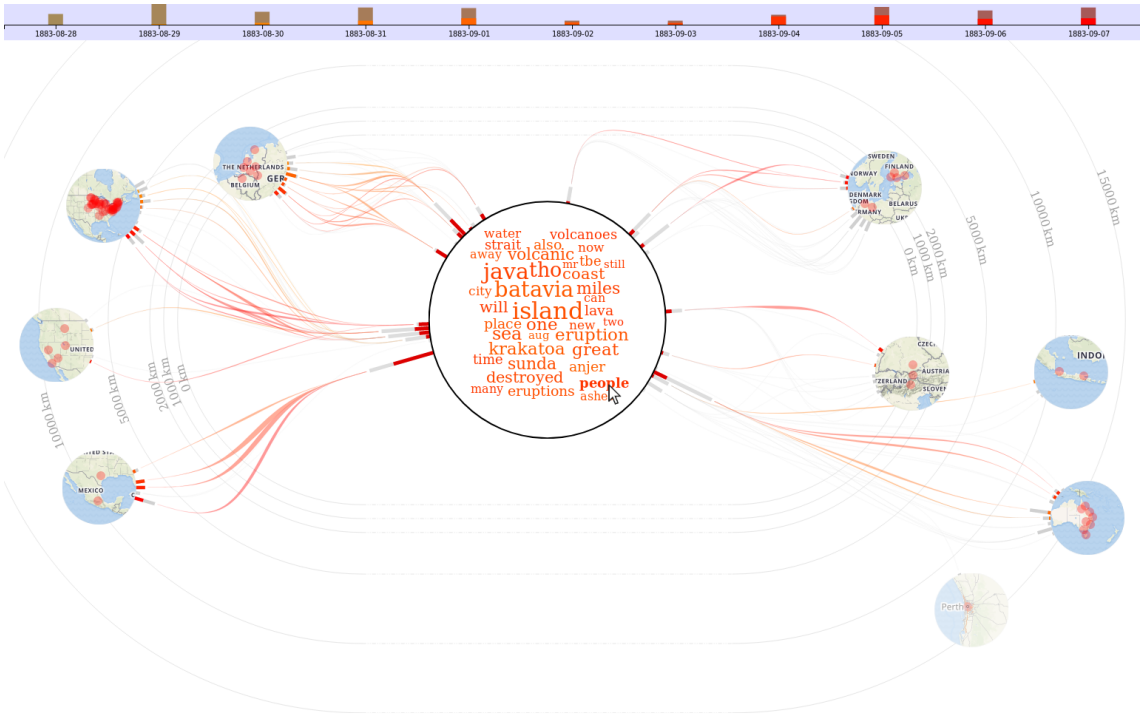


(d) Western Australia inset selected.



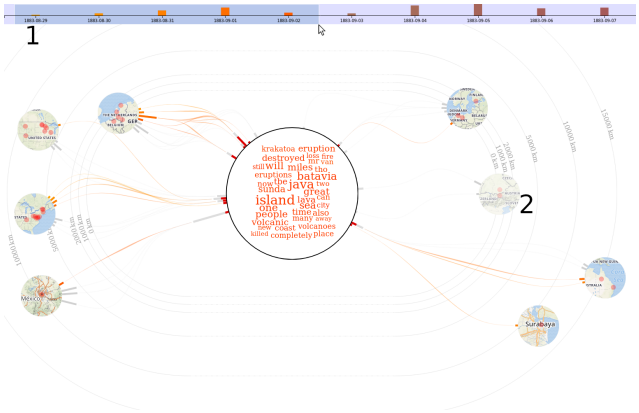
(e) Comparison of linking when hovering the word '*coast*'.

Figure 6.1: Screenshots of an usage example, where mentions of the word '*coast*' in the USA and Australia is compared. First, the directional histogram bars around the word cloud connecting to map insets of those locations are identified in (a–d). Then, the word is brushed in the word cloud in (e), and the linked parts of the histogram bars are compared.

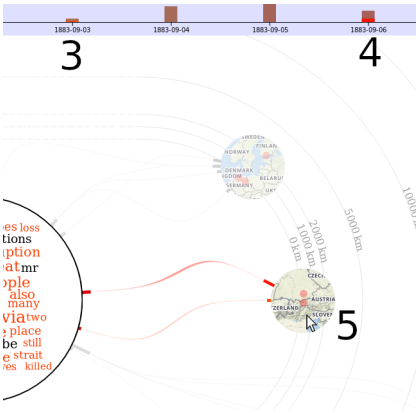


(a) Hovering over the word '*people*'.

(b) Date histogram bar for the drilled down dataset of articles with the word '*people*'.



(c) First half of time period selected.



(d) Hovering a map inset.

Figure 6.2: Screenshots of an usage example, where we explore the temporal distribution of the use of the word '*people*'. First, the word is located and hovered (a). Then, the dataset is drilled down (b). First, we explore the geographical distribution for the first half of the time period by brushing the date histogram (c), then we look at the temporal distribution for one inset (d).

6.1.2 Analysing Temporal Distribution of Word Usage

In the second usage example, we want to explore the temporal distribution of the use of the word '*people*'. From the initial view, we locate the word in the word cloud. We then hover over the word, as shown in Figure 6.2a. From the linking in the date histogram, we can now already compare the word use each day to the total amount of documents for that day. This already indicates that the word was used more towards the end of the time period.

To further explore this, we click on the word, thereby drilling down into the dataset. Now, only documents that contain the word '*people*' are visible. We now see the temporal distribution of the word use in the date histogram's total, as seen in Figure 6.2b.

We select the first five days of the time period in the date histogram in Figure 6.2c (1). This links documents published in that timeframe throughout the visualisation, and we notice that the map inset showing Tirol and Munich is not linked. In Figure 6.2d, we then hover the mouse over that inset (5) and then find out from the date histogram bar (3, 4) that the word '*people*' was first used in those locations on the 3rd and then the 6th of September, 1883, both in the second half of the dataset's time range.

6.1.3 Finding Documents Based on Keywords, Geographic or Temporal Constraints

In the third usage example, we want to demonstrate a drill down task. For the given dataset, we want to find all documents which

1. were published in Europe.
2. mention '*Krakatoa*' and '*Sunda*'.
3. were published in August of 1883.

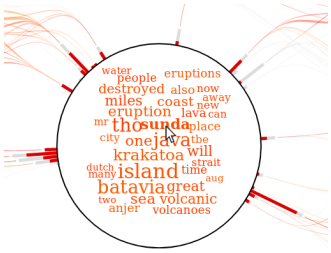
There are multiple orders in which the drill down could be done. We start by filtering out all documents which do not contain the two words. From the initial view, we locate '*Krakatoa*' in the word cloud (Figure 6.3a), move the mouse over the word and click it. We then repeat the same procedure for the word '*Sunda*', as shown in Figure 6.3b. In Figure 6.3c, we then select the days belonging to August in the date histogram, which further drills down the dataset.

The resulting visualisation is shown in Figure 6.3d. It shows seven map insets, of which only three—the Netherlands (1), Berlin (2) and Tirol (3)—are in Europe. With the current prototype, it is not possible to drill down into a union selection, so we instead sequentially select the insets.

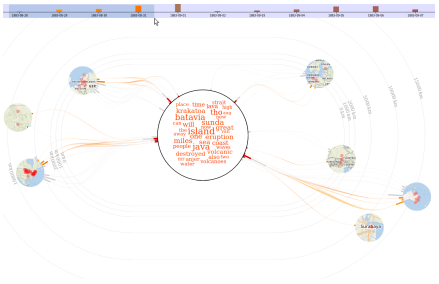
We first click on the map inset showing the Netherlands (1). The documents that are still part of the visualisation are listed in the document list shown in Figure 6.3e. We can now open single documents from that list in new windows. Using the back button, we return to the previous subset of data, which is visualised in Figure 6.3d. We then select the Berlin map inset (2), drilling down into that. This results in the data shown in Figure 6.3f. Going back once more and then selecting the Tirol map inset (3), we get the list of documents for that location in Figure 6.3g.

This usage example shows us that the hierarchical clustering of locations can still be a bit impractical, depending on how the locations were clustered. In Section 8.2, we discuss some possibilities to improve the handling.

(a) Hovering '*Krakatoa*'.



(b) Hovering '*Sunda*'.



(c) Selecting the date range.

(d) Drilled-down view.

8 articles visible

Date	Title (Newspaper)	Location
August 31, 1883	Leeuwarder Courant	Leeuwarden
August 31, 1883	De standaard	Amsterdam
August 31, 1883	De standaard	Amsterdam
August 31, 1883	Algemeen Handelsblad	Amsterdam
August 31, 1883	Het nieuws van den dag : kleine courant	Amsterdam
August 31, 1883	De Tijd : godsdienstig-staatkundig dagblad	s-Hertogenbosch
August 30, 1883	De standaard	Amsterdam
August 30, 1883	Algemeen Handelsblad	Amsterdam

(e) Document list for Netherlands map inset.

3 articles visible

Date	Title (Newspaper)	Location
August 30, 1883	Berliner Börsenzeitung (Morgenausgabe)	Berlin
August 30, 1883	Berliner Tageblatt	Berlin
August 28, 1883	Berliner Tageblatt	Berlin

(f) Document list for Berlin map inset.

1 article visible

Date	Title (Newspaper)	Location
August 31, 1883	Bote für Tirol	Tirol

(g) Document list for Tirol map inset.

Figure 6.3: Screenshots of an usage example, where we drill down into a dataset using multiple criteria. First, we reduce the dataset to documents containing the word '*Krakatoa*' (a). Then, we reduce it to documents also containing the word '*Sunda*' (b). After that, we reduce the date range to only show documents from August 1883 (c), resulting in a subset of documents clustered into seven map insets (d). For the three map insets located in Europe, we then explore which documents are contained (e–g).

6.2 Expert Review

We have conducted an expert review together with a postgraduate at the Institute of Literary Studies at the University of Stuttgart. The test subject had been involved in creating the Krakatoa test dataset presented in Section 5.3.2. She is also working on the Oceanic Exchanges [13] project. This makes her a potential user of the prototype, and thus we wanted to let her try it out and collect her feedback. The first objective was to observe her use of the prototype in order to find potential usability flaws and unintuitive parts of the visualisation. The second objective was to gather useful feedback and suggestions for improvement and additional features. For this, the opinion of a stakeholder and a non-programmer has been very helpful.

For the review, the prototype's version *v1.0*—introduced in Section 4.2.6—was hosted and opened on a laptop running Arch Linux with the 4.18.5 Linux kernel. The server back end was self-hosted using Flask, and the web page was opened in Mozilla Firefox version 61.0. The laptop had a 7^{th} generation Intel i7 CPU, 16GB of RAM and a 1920×1080 pixel $14''$ (35.6 cm) display. The test subject was given the laptop with the browser window open and a mouse connected. For documentation purposes, audio of the interview and a screen capture of the laptop's monitor was recorded with the consent of the test subject.

We first explained the format of the expert review to the test subject and let her sign the declaration of consent. We then briefly demonstrated the prototype's capabilities and all parts of the visualisation. All questions raised by the test subject during that phase were answered.

The test subject was then given four questions to answer about the dataset. Her actions while finding the answers were observed, and possible questions answered. When it was apparent that the test subject was stuck, hints were given to lead her back on the right track.

As the test subject has not been part of the working group exploring the dataset, it was not possible to compare her use of the visualisation to her previous working routine. This would have been a useful insight into the shortcomings the prototype can bridge, and possible additional features that could be added. Instead of this comparison, we discussed how she could see the prototype helping her in her work. We also got some helpful pointers as to what was still missing. Finally, we let the test subject fill out a questionnaire, providing some additional feedback.

6.2.1 Observations

We notice a learning curve with the test subject. Initially, she needed to take her time to navigate the prototype. She also only used the date histogram at the top and the word cloud in the centre in the beginning. After using the prototype for a while and getting a few hints on the use of other parts of the visualisation, she used more parts of the prototype, and her actions became faster and more confident.

The first task to solve was '*Find a word which is not used in newspapers in the western United States*'. The test subject, after having located the map inset containing all locations in the western United States, started a brute force search through the words in the word cloud until she had found a word which did not brush that inset. That took a while, partly because the smooth animations in the brushing and linking, combined with fast mouse movements in the word cloud, led to the test subject not noticing the highlights, or noticing them too late. Eventually, a word was however found.

The second task was '*In which locations were most articles published in the first few days and where were did the articles mostly published towards the end of the observed period?*'. For the first part of the task, it showed that the test subject was not yet acquainted with the prototype. Initially, she brute-forced by brushing single dates in the date histogram at the top. Then, she drilled down into single countries by clicking the insets, and then comparing the date histograms at the top. This way, the context switches made direct comparisons hard. After giving the test subject a hint, she instead hovered the map insets and compared the brushed date histograms at the top. She also directly compared the radial date histograms around the map insets. This way, she found the answer without problems. For the second part of the question, she found the answer in short time, showing that her previous struggles were primarily because of her lack of experience with the prototype.

The third task was '*Find all articles published in Finland mentioning "Bantam"*'. By now, the test subject was acquainted with the prototype and solved the task very fast. First, she drilled down into the dataset until only articles published in Finland were visible by clicking on the map insets. Then, she located the word 'Bantam' in the word cloud and hovered it. The test subject then pointed out that the highlighted six articles in the document list were the result, which was correct. However, she could also have clicked the word in the word cloud, which would have made the selection permanent and allowed her to explore the individual articles. This is just a side note, the required task was solved well and quickly.

The last task was '*Find all articles published between the 1st and 5th of September, 1883, which were published in Mexico City and do not mention earthquakes*'. The test subject also solved this task very fast. She first reduced the dataset to show only articles published in Mexico City by clicking on the map insets. She then reduced the date range by selecting it in the date histogram at the top. Finally, she found the word 'earthquake' in the word cloud and hovered it. She then pointed out the two articles which were *not* brushed in the document list, which were the correct ones. Her strategy shows that there are multiple ways to solve such tasks with the prototype, as we ourselves first reduced the date range in our test run.

6.2.2 User Feedback

The feedback to the visualisation has been largely positive, and only minor flaws were pointed out. However, the test subject has some useful propositions on how to further enhance the prototype. The test subject especially likes the highly integrated visualisation and '*the possibility to have everything relevant available at first glance without being overwhelmed by too much input*'. Having an interface which gives a good summary overview without temporal separation has been a desired goal from the start, and so this piece of feedback is very welcome. The test subject also likes the single document view and that the original scans of the newspapers in the dataset can be directly accessed from it.

A point of critique is that the word cloud is monolingual although the corpus is not. This insight is surprising for us: Making the word cloud monolingual has been a practical decision, made to actually show the most used words. For a researcher in Literary Studies however, the untranslated use of words seems to be more interesting. This might possibly require different weighting of terms in the word cloud, such that insights can properly be gained from the visualisation. The test subject explains that location names in particular, but also certain phrases, are different in different languages. She brings up the example of the volcano's name, 'Krakatau' in the original

Indonesian, which is written ‘Krakatoa’ in English. In the time directly following the eruption, German newspapers would still use the correct term, but switch to the incorrect spelling used by newspapers in other countries after a while. By homogenising the language of the word cloud, such subtle interactions are lost.

Another point of critique is the colour scale used for encoding the dates. The gradient from red to orange used in the latest versions of the prototype introduced in Sections 4.2.5 to 4.2.6 is deemed too aggressive. The test subject states she intentionally looked away from parts of the visualisation after a while because they are too aggressively coloured, and looking at them for a longer time is straining. This is of course an undesired effect that hinders productive use of the prototype, and so we are grateful for that piece of feedback.

When asked how the prototype would facilitate her work with the dataset, the test subject answered that it helps her see the amount of publication in the respective country at first glance. She also confirms that she would use the prototype in her daily work, specifically to look at datasets describing single events.

The test subject also confirms that she found the directional layout of the map insets intuitive. To explore this, she was shown a second version of the prototype that used great circle direction instead of equirectangular direction for positioning the map insets. The test subject however interjected that her bias towards the first version might be because she got acquainted with that one first. To objectively compare the two versions, A/B testing with more test subjects is required. This could not be evaluated using the expert review. To underline that the directional layout was intuitive, the test subject was also asked whether she missed the presence of a world map as the centre of the visualisation, which she stated she did not. The test subject also stated that she found the brushing and linking and the interactions with the visualisation intuitive and helpful.

On the point of layouts, we also had a longer discussion about the choice of map origin. For the single document version introduced in Section 4.1, we have chosen the document’s publication location as origin, as that seems sensible. For the document collection visualisations introduced in Section 4.2, this is not an option as there are multiple publication locations. Thus, we choose the user’s own location as origin, with the rationale that that fits best into the user’s world view and makes the layout of the map insets as intuitive as possible. The test subject however states that for this dataset, she would have found the topic’s main location—the island of Krakatoa—to be the most intuitive origin. However, this solution might not fit every dataset, as some datasets do not have one singular main location. We conclude that the choice of origin should be left to the user and easily configurable in the interface.

We also discussed future improvements to the visualisation. The test subjects research revolves around finding out how newspaper articles of that time were spread and reprinted, and therefore how news spread. This also involves the spread of misinformation and information that is suppressed in certain countries or regions, i.e. does not spread. One proposition the test subject had is to add a search functionality that highlights and isolates documents that mention a certain term that the user provides. This is currently only possible for the terms visible in the word cloud and would be a good improvement. She also mentioned that, given a dataset that already contains the reprint information, visualising the spreading patterns of news in the prototype would be a good addition. A last proposition is to add better support to compare selections side-by-side.

7 Discussion

In this chapter, we discuss some of the decisions made in the creation of the prototypes, and what the alternatives would have been.

7.1 Forgoing a Central Map

One core feature of our prototypes is that they do not use a map as central part of the visualisation despite visualising data with geographical references. Inspired by visual cues used in several works [4, 17, 22] on the indication of off-screen elements and by the work of Brodkorb et al. [8], we created a visualisation where clusters of geolocations are indicated by small map insets. Forgoing a map as the central component means creating discontinuities, which in turn makes it harder for the user to spatially relate the insets to each other. In order to make it easier for the user to create a mental image of the geographical layout, we consider many possibilities regarding the layout of the visualisation and the placement of the insets. This also raises new questions, which are discussed in Sections 7.2 and 7.3.

However, forgoing a central map means that large parts of the visualisation can be freed up. This allows us to include the directional histogram, the radial date histograms around the map insets and the splines connecting the two. During the expert review, the test subject stated that she did not miss the central map and that she found the visualisation design we created useful. This should be confirmed with a larger study in the future.

The test subject also stated that she oriented herself in the visualisation using country shapes and coastlines. This indicates that the hierarchical clustering could be enhanced by respecting continent, country and state borders. Modifying the map insets to more clearly showing country borders—or even making the insets country-shaped—should also improve the readability.

Last, indicating a silhouette world map where the isolines are currently drawn could also give a useful visual cue. However, this silhouette would be quite distorted, and because of the final force layout step, map insets are not guaranteed to be in their originally projected position, although the silhouette could be further distorted based on the movement of the map insets. This merits further exploration in the future.

7.2 Choice of Forward Azimuth

For the initial prototypes, we use the great circle direction's forward azimuth to place the map insets. The goal is to place the insets as accurately as possible, and the great circle direction accomplishes that. However, when testing this with the Krakatoa dataset, the great circle distance proved to yield

unintuitive results. One example is the placement of the Surabaya map inset (E) in Figure 4.3c, where it is placed north of New York City, despite being located as far south as 7° 15′ northern latitude.

We conclude that people have had their intuitive view of the world shaped by the Mercator projection, as indicated by Gudmundsson and Alerstam [21], and that placing the insets accordingly should yield much more intuitive results. As calculations with the Mercator projection include integration—whilst calculating direction with the equirectangular projection only requires a call to `atan2`—we choose to use equirectangular projection instead. The errors created that way are negligible for coordinates between 60° southern latitude and 60° northern latitude. This range includes most densely populated areas on Earth, and all publication locations used in the Krakatoa dataset. For future work and datasets containing locations farther north or south, the performance hit when using Mercator should be weighted against the loss in precision from using equirectangular projection.

7.3 Disrupting the Projection Space

In the single document prototype introduced in Section 4.1, a vertical gap is left in the projection space as indicated in Figure 4.2. The reason here is to allow the connector splines to exit the side of the text view horizontally without creating too much tension in the splines. It also fits the screen aspect ratio of 16:9 better, which leaves only little space above and below the text view. However, it skews the placement of the insets, which might be confusing to the user.

For the subsequent prototypes in Section 4.2, spline tension is no longer an issue, but the aspect ratio of today's displays still is. On a screen where the width is nearly twice the height, projecting something radially becomes a problem. There are essentially three ways to solve this:

1. **Use different scaling factors for vertical and horizontal scaling.**
   This distorts not only the distances, but also the angles. The result is a very undesirable projection that is very easy to misread.
2. **Use the horizontal scaling factors for vertical and horizontal scaling and hope that vertical distances are at the most half as large as horizontal ones.**
   This approach is a bit optimistic, as it assumes that the horizontal distances are on average longer than the vertical ones, i.e. the projected-from space also has a 16:9 aspect ratio. For some configurations of our prototype, depending on the origin, this might actually work. A world map in equirectangular projection has an aspect ratio of exactly 2:1.
3. **Use the vertical scaling factors so that the projection fits.**
   This is the safest method, but it generates large bits of unused space left and right of the—essentially square—projection area.

While the second alternative can be viable—at least for a origin near the equator—it is too optimistic an approach to be generalised. We therefore choose the third alternative. This does not distort the scaling. By dividing up the projection space into two hemispheres, the unused space can be used to display the word cloud in the middle.

Of course, splitting up the projection area distorts the angles as well. In order to reduce the effect of the distortion on the user, visual cues are added by drawing isolines. Those indicate the distance from the origin, and map insets are placed at their respective distance, which is also respected by the force layout step at the end. In the middle, where the projection area is split, the isolines are drawn using dashed lines to indicate that this is not part of the projection area. The isolines and dashed lines can for instance be seen in Figure 4.10. The isolines also help by showing where the zero line is, which has a observable offset.

Another possibility would be to remove the centre part of the visualisation. This would drastically reduce the offset of the zero point, and leave enough space at the top and bottom. However, the centre part of the visualisation is essential, as it contains the word cloud and the directional histogram. All in all, we are confident our solution is near-optimal for the problems we aim to solve.

7.4 Colour Choice

The expert review showed us that our choice of colour for the dates is not ideal. The test subject stated that the red and orange is too aggressive and difficult to look at. The intuitive solution is to find a different part of the colour spectrum—for example a darker cyan to dark blue—and use that for the colour scale. However, the interpretation of colours might be culture-specific, or even specific to an individual, and choosing any other colour scale might lead to similar problems with another user. In order to find better colours, a larger user study is necessary.

The solution is to provide sensible defaults for the colour scale based on our current feedback from users, but leaving it to the user to modify the colour scale. This means providing an accessible setting for start and end colour.

7.5 Multilingual Word Cloud

The expert review also revealed that translating the texts before creating the word cloud actually removed information that the test subject would have been interested in, such as differences in location name spelling. Creating a multilingual word cloud from the original texts might reveal such information. However, this would also increase the amount of unique words that could be part of the word cloud, and the weighting of the words should be revisited.

One issue might be that the distribution over the languages used in the documents might not be uniform. Words in some languages might therefore be suppressed from the word cloud because they only appear in very few documents. One remedy would be to use a weighting based on *tf-idf* instead, which would highlight words used often, but in few documents. However, this might be misinterpreted. Another possibility is to weigh the words inversely relative to the amount of documents in that language in order to normalise the language distributions.

Both possibilities skew the ratios and invite misinterpretation. With some further preprocessing of the text, different spellings and translations of the same word could be identified as such, and this information could be incorporated into the visualisation. This could mean just grouping the different versions of the word in one place, or showing one and splitting it up on hover, or displaying a small pie chart beside the word.

7.6 Creating a Useful Visualisation for Specialists in American Studies

The goal of this work has been to provide a visualisation for specialists in American studies to explore datasets of documents with temporal and geographical properties. Based on the feedback from the expert review, we have succeeded in doing so. However, a larger group of specialists should be consulted to get further and more reliable feedback.

We conclude that our prototype can help specialists to explore such datasets and gain insight on the data. However, there are still some usability issues and open questions, which we discuss in Sections 7.1 to 7.5. Besides getting more feedback from different users, the main improvement is to provide customisation points. It would also help to provide tooltip help or a user guide.

8 Conclusions and Future Work

8.1 Conclusions

With this work, we create a visualisation prototype to indicate geographical references within a text to the user. We also create a visualisation prototype that can visualise the geographical and temporal distribution of a collection of documents. With the latter, we manage over multiple iterations to find flaws in the visualisation and subsequently solve them. The final version presents itself as a highly integrated, interactive visualisation that supports brushing and linking interaction, drill down exploration into the dataset and an undo functionality. We also manage to create a visualisation that intuitively shows geographical distribution without requiring a full-scale map as central component.

The prototype was received positively by a test subject, who after a short familiarisation was able to answer complex questions about the dataset. By creating usage examples and discussing the prototype with the test subject, we have also been able to identify some weak points of the prototype, which can be addressed in future versions.

After having discussed in depth the choice of colours and the origin location, it is clear that those should remain configurable by the user. For our test subject in Section 6.2, the red-to-orange colour coding of the dates was deemed too aggressive. However, we suspect that the interpretation of colours is dependent of culture and personal preference, and conclude that the colour scale for the dates should be configurable. As for the origin location for the inset layout, both the user's location and the epicentre of the dataset's main event have been discussed as viable alternatives. The first one however is dependent on the user's world view, as for example someone born and raised in Germany but currently living in the United States might still want Germany as origin. Regarding the latter, not every dataset might have one distinct epicentre, or a consensus on where the epicentre is. We therefore propose letting users select the origin themselves.

We also noticed some weaknesses in the selection of subdatasets. Apart from the date histogram, the user is not able to select a range of objects. Adding a possibility to first build up a collection and then doing the drill down would yield even better control over the selection and exploration. During the expert review, it also became clear that some of the histogram bars were too short to read the highlighted fractions.

In conclusion, the prototype is a success, and the goal of creating a visualisation of documents with geographical references without basing it on a map has been reached. However, there are still improvements to be made and changes to be explored in future work.

8.2 Future Work

Through the expert review, we have gathered valuable feedback on what a user of the prototype would like to be able to do. One point of critique is the monolingual word cloud, so future work could include exploring ways to create a multilingual word cloud. We have already discussed some possibilities in Section 7.5, which could be experimentally implemented. Another point is a free search feature, where arbitrary terms or phrases could be entered. The terms could then either be added to the word cloud or placed in some other position, and would be part of the brushing and linking interaction in every way. This would allow more specific exploration of the dataset and facilitate the search for spelling variants. Currently, only brushing of words that are part of the word cloud is possible.

For the task of finding and tracing reprints, some further improvements could be implemented. One example would be visually linking reprints, for example via directed lines connecting locations and dates in the time line. The techniques used by Cao et al. on *Whisper* [10] could be taken as inspiration. *Whisper* uses lines with markers to visualise retweets. The reprinting could also influence the weighting of articles and words in the word cloud, as the reprinted article could be considered as not having full weight. Doing this would however require the dataset to be augmented with information regarding which documents are reprints of which.

The concept of navigating through a graph where nodes with a certain similarity are connected by edges that we introduce in Section 4.2.1 could also be further investigated. We have not continued using it in later version of the prototype, but the concept has potential to succeed if done right. For that, fitting distance functions have to be found and the visualisation improved. This way, an interactive visualisation for exploration of similar documents could be created. A good distance function could utilise not only geographical and temporal distance, but also topic similarity and even the reprint metadata mentioned above. However, the development of a good and general distance function is hard. It also would require further preprocessing of the data.

Another improvement that would require preprocessing of the dataset would be to also visualise sentiment of the documents. This would allow the researchers to explore how sentiment about some event differs at different locations and different points in time. Together with the tracing of reprints and suppressed news, this could lead to interesting insights on cultural and political interrelations.

Being able to save a subselection of documents in a selection object would also facilitate exploration and comparison. This could be realised in the user interface by pressing a key when in a brushing selection, and an item then appears in a list of selections somewhere at an edge. This selection could then be brushed and would behave as any other part of the visualisation. A stored selection bar would also be a good spot to place the arbitrary terms proposed above. In that vein, making it possible to do union selections should also be considered. As of now, only intersection selections are possible by drilling down into subsets of the data. For example, selecting multiple map insets and drilling down into that selection would make the drill down exploration more flexible. Such union selections should then also be possible to save in the stored selection bar.

Another point of improvement would be to supplement the visual cues in the visualisation. Currently, the distance isolines are the only cues present. Those isolines could be supplemented with the silhouette coastlines of the landmasses located at those locations, and country borders as well. The coastlines would of course have to be distorted following not only the projection, but also the force-directed displacement of the map insets, such that the map inset would be placed exactly

above its location in the silhouette world map. The risk here is that the world map is too distorted to actually recognise the shapes of countries, but it could be worth a try, as it would help the user's mental model in setting the map insets in perspective. Adding some lines for geographic direction, for example in 15° steps, would also help, albeit those would also have to be distorted in the final force layout step.

Improving the visual cues also includes the histogram bars. The expert review revealed that it is hard to get a reading on the fractions of currently linked articles in the histogram bars, especially for low amounts of brushing. The test subject remarked that she would have liked percentage numbers displayed. While we think that adding more text to the visualisation could increase visual clutter, the indication of low percentages in those bars should be improved. A simple first fix would be to show a well-perceivable small-sized bar, even for fractions under a threshold, so that it is guaranteed to be visible. Of course, that would increase the lie factor for those bars. The possibilities regarding this should be explored in detail and weighed against their drawbacks.

Currently missing in the prototype is the possibility to compare two selections. Comparing side-by-side (spatial separation) would reduce the size of the visualisation, which might make it unreadable. Having the possibility of storing selections as discussed above, temporal separation could be used to compare selections. Another possibility would be to display both selections in the same visualisation at the same time using different colour scales. Of course, this again means having to worry about the user's subjective interpretation of the colours. It might also overload the visualisation. Those issues should be discussed with user interface experts and tried out in a user study.

It turned out during development and the expert review that there are several configuration points of the visualisation which should be left to the user. This includes the type of projection used for placing the map insets, the date's colour scale and the choice of origin. In future, the visualisation should provide a way to change these settings that is easy to access and understand. The changed settings should also instantly affect the visualisation, so that reloading is not necessary. The defaults for those settings should be evaluated during a user study with more users. Especially for the choice of map projection for the placement, a larger study could produce interesting and useful results.

For better usability, it should also in future be possible to save one's progress and settings, to share data subsets with others, to upload new data and download subsets of the data. For this, the server back end and the REST interface have to be extended. In future, it could be possible to feed a totally new dataset into the interface and start to explore it. With the visualisation so far progressed, it might also be time to move the implementation to a server accessible from the internet—in a password-protected fashion. This would allow us to let more members of the Oceanic Exchanges program test it, which might also give us more useful feedback with more variety of test subjects.

Further interesting possibilities come to mind if the datasets are preprocessed using named entity recognition (NER). Using NER, location names within the texts can be found and internally marked, similar to the markup used for the first prototype in Section 4.1. For one, the single document view introduced to the prototype in the last version in Section 4.2.6 could be improved to work like the single document prototype in Section 4.1, visualising the location references within the text. Additionally, relations between publication locations and mentioned locations could also be visualised in the main view. This again has the potential to make the main visualisation too crowded, but could also provide further insight on the dataset. The visualisation choices for this should be investigated in detail.

Future work should also include exploring the scalability of the visualisation with larger datasets. Furthermore, datasets that are not as focused on a single topic as the Krakatoa dataset should be tried out. This dataset only contains newspaper articles about the eruption, and so the density of the desired information is unnaturally high. It should be interesting to explore how the visualisation fares with corpora containing for one many more documents and for the other a higher diversity of topics. One risk when developing a program is that it is too much custom-engineered to fit one use case or dataset. Looking at different datasets with different characteristics will most probably reveal further ways to improve the current visualisation concept to make the visualisation more flexible and usable.

Bibliography

Literature

[1] S. Ahern, M. Naaman, R. Nair, J. H.-I. Yang. 'World Explorer'. In: *Proceedings of the 2007 Conference on Digital Libraries*. ACM Press, 2007. DOI: 10.1145/1255175.1255177 (cit. on pp. 23, 24, 35).

[2] P. Baudisch, N. Good, V. Bellotti, P. Schraedley. 'Keeping Things in Context'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '02*. ACM Press, 2002. DOI: 10.1145/503376.503423 (cit. on p. 21).

[3] P. Baudisch, N. Good, P. Stewart. 'Focus Plus Context Screens'. In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology - UIST '01*. ACM Press, 2001. DOI: 10.1145/502348.502354 (cit. on p. 21).

[4] P. Baudisch, R. Rosenholtz. 'Halo: A Technique for Visualizing Off-Screen Objects'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2003, pp. 481–488. DOI: 10.1145/642611.642695 (cit. on pp. 22, 57).

[5] C. de Boor. *A Practical Guide to Splines*. Springer New York, 1978. DOI: 10.1007/978-1-4612-6333-3 (cit. on p. 12).

[6] H. Bosch, D. Thom, M. Worner, S. Koch, E. Puttmann, D. Jackle, T. Ertl. 'ScatterBlogs: Geo-Spatial Document Analysis'. In: *Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, Oct. 2011. DOI: 10.1109/vast.2011.6102488 (cit. on pp. 23, 24, 35).

[7] R. S. Bradley. 'The Explosive Volcanic Eruption Signal in Northern Hemisphere Continental Temperature Records'. In: *Climatic Change* 12.3 (June 1988), pp. 221–243. DOI: 10.1007/bf00139431 (cit. on p. 47).

[8] F. Brodkorb, A. Kuijper, G. Andrienko, N. Andrienko, T. Von Landesberger. 'Overview with Details for Exploring Geo-Located Graphs on Maps'. In: *Information Visualization* 15.3 (2016), pp. 214–237. DOI: 10.1177/1473871615597077 (cit. on pp. 23, 57).

[9] A. Buja, J. A. McDonald, J. Michalak, W. Stuetzle. 'Interactive Data Visualization Using Focusing and Linking'. In: *Proceedings of the 1991 IEEE Conference on Visualization - '91*. IEEE Comput. Soc. Press, 1991. DOI: 10.1109/visual.1991.175794 (cit. on p. 14).

[10] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, H. Qu. 'Whisper: Tracing the Spatiotemporal Process of Information Diffusion in Real Time'. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2649–2658. DOI: 10.1109/tvcg.2012.291 (cit. on pp. 25, 27, 28, 62).

[11] I. Cho, W. Dou, D. X. Wang, E. Sauda, W. Ribarsky. 'VAiRoma: A Visual Analytics System for Making Sense of Places, Times, and Events in Roman History'. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 210–219. DOI: 10.1109/tvcg.2015.2467971 (cit. on pp. 25, 26).

[12] A. Cockburn, A. Karlson, B. B. Bederson. 'A Review of Overview+Detail, Zooming, and Focus+Context Interfaces'. In: *ACM Computing Surveys (CSUR)* 41.1 (2009), p. 2. DOI: 10.1145/1456650.1456652 (cit. on pp. 21, 22).

[13] R. Cordell, M. Beals, I. Galina, M. Priewe, E. Priani, H. Salmi, J. Verheul, R. Alegre, S. Koch, T. Hauswedell, P. Fyfe, J. Hetherington, E. Lorang, A. Nivala, S. Pado, L.-K. Soh, M. Terras. 'Oceanic Exchanges'. 2017 (cit. on pp. 9, 54).

[14] G. Draper, Y. Livnat, R. Riesenfeld. 'A Survey of Radial Methods for Information Visualization'. In: *IEEE Transactions on Visualization and Computer Graphics* 15.5 (Sept. 2009), pp. 759–776. DOI: 10.1109/tvcg.2009.23 (cit. on p. 25).

[15] Y. Drocourt, R. Borgo, K. Scharrer, T. Murray, S. I. Bevan, M. Chen. 'Temporal Visualization of Boundary-Based Geo-Information Using Radial Projection'. In: *Computer Graphics Forum* 30.3 (June 2011), pp. 981–990. DOI: 10.1111/j.1467-8659.2011.01947.x (cit. on pp. 25, 27, 28).

[16] B. Dumas, T. Broché, L. Hoste, B. Signer. 'ViDaX'. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*. ACM Press, 2012. DOI: 10.1145/2254556.2254702 (cit. on pp. 27, 28).

[17] M. Frisch, R. Dachselt. 'Visualizing Offscreen Elements of Node-Link Diagrams'. In: *Information Visualization* 12.2 (2013), pp. 133–162. DOI: 10.1177/1473871612473589 (cit. on pp. 23, 57).

[18] G. W. Furnas. 'Generalized Fisheye Views'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '86*. ACM Press, 1986. DOI: 10.1145/22627.22342 (cit. on pp. 21, 36).

[19] S. Ghani, N. H. Riche, N. Elmqvist. 'Dynamic Insets for Context-Aware Graph Navigation'. In: *Computer Graphics Forum* 30.3 (June 2011), pp. 861–870. DOI: 10.1111/j.1467-8659.2011.01935.x (cit. on p. 23).

[20] I. N. Gregory, A. Hardie. 'Visual GISting: Bringing Together Corpus Linguistics and Geographical Information Systems'. In: *Literary and Linguistic Computing* 26.3 (May 2011), pp. 297–314. DOI: 10.1093/llc/fqr022 (cit. on p. 24).

[21] G. A. Gudmundsson, T. Alerstam. 'Optimal Map Projections for Analysing Long-Distance Migration Routes'. In: *Journal of Avian Biology* 29.4 (1998), pp. 597–605. ISSN: 09088857, 1600048X. URL: http://www.jstor.org/stable/3677180 (cit. on pp. 17, 58).

[22] S. Gustafson, P. Baudisch, C. Gutwin, P. Irani. 'Wedge: Clutter-Free Visualization of Off-Screen Locations'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2008, pp. 787–796. DOI: 10.1145/1357054.1357179 (cit. on pp. 22, 23, 57).

[23] J. Harris. 'Word Clouds Considered Harmful'. In: *Nieman Journalism Lab* (2011) (cit. on p. 14).

[24] J. He, C. Chen. 'Spatiotemporal Analytics of Topic Trajectory'. In: *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction - VINCI '16*. ACM Press, 2016. DOI: 10.1145/2968220.2968244 (cit. on pp. 27, 28).

[25] A. Jaffe, M. Naaman, T. Tassa, M. Davis. 'Generating Summaries and Visualization for Large Collections of Geo-Referenced Photographs'. In: *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval - MIR '06*. ACM Press, 2006. DOI: 10.1145/1178677.1178692 (cit. on pp. 23, 35).

[26] D. Keim. 'Information Visualization and Visual Data Mining'. In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8. DOI: 10.1109/2945.981847 (cit. on p. 15).

[27] Y. K. Leung, M. D. Apperley. 'A Review and Taxonomy of Distortion-Oriented Presentation Techniques'. In: *ACM Transactions on Computer-Human Interaction* 1.2 (June 1994), pp. 126–160. DOI: 10.1145/180171.180173 (cit. on p. 22).

[28] J. D. Mackinlay, G. G. Robertson, S. K. Card. 'The Perspective Wall'. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '91*. ACM Press, 1991. DOI: 10.1145/108844.108870 (cit. on pp. 21, 22).

[29] M. McGill, G. Salton. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983 (cit. on p. 13).

[30] A. Mehler, Y. Bao, X. Li, Y. Wang, S. Skiena. 'Spatial Analysis of News Sources'. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (Sept. 2006), pp. 765–772. DOI: 10.1109/tvcg.2006.179 (cit. on p. 24).

[31] G. G. Robertson, J. D. Mackinlay. 'The Document Lens'. In: *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology - UIST '93*. ACM Press, 1993. DOI: 10.1145/168642.168652 (cit. on p. 22).

[32] R. Rosenholtz, Y. Li, L. Nakano. 'Measuring Visual Clutter'. In: *Journal of Vision* 7.2 (2007), pp. 17–17 (cit. on pp. 23, 34).

[33] B. Shneiderman. 'The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations'. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. IEEE Comput. Soc. Press, 1996. DOI: 10.1109/vl.1996.545307 (cit. on p. 15).

[34] B. Shneiderman, C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Englisch. 4th ed. Boston, Mass.: Pearson/Addison-Wesley, 2005, XVIII, 652 Seiten. ISBN: 0-321-26978-0. URL: http://www.gbv.de/dms/ilmenau/toc/386446083.PDF (cit. on pp. 9, 38).

[35] H. A. Simmons, G. D. Gore. *Plane and Spherical Trigonometry*. John Wiley & Sons, 1945 (cit. on p. 17).

[36] J. P. Snyder. *Flattening the Earth: Two Thousand Years of Map Projections*. University of Chicago Press, 1997 (cit. on pp. 15, 17).

[37] J. Stasko, E. Zhang. 'Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations'. In: *Proceedings of the 2000 IEEE Symposium on Information Visualization*. IEEE Comput. Soc, 2000. DOI: 10.1109/infvis.2000.885091 (cit. on pp. 25, 28).

[38] D. F. Swayne, D. T. Lang, A. Buja, D. Cook. 'GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization'. In: *Computational Statistics & Data Analysis* 43.4 (Aug. 2003), pp. 423–444. DOI: 10.1016/s0167-9473(02)00286-4 (cit. on p. 15).

[39] G. J. Symons, J. W. Judd, S. R. Strachey, W. J. L. Wharton, F. J. Evans, F. A. R. Russell, D. Archibald, G. M. Whipple. *The Eruption of Krakatoa: And Subsequent Phenomena*. Trübner & Company, 1888 (cit. on p. 47).

[40] W. R. Tobler. 'A Computer Movie Simulating Urban Growth in the Detroit Region'. In: *Economic Geography* 46 (June 1970), p. 234. DOI: 10.2307/143141 (cit. on pp. 9, 34).

[41] E. R. Tufte. *The Visual Display of Quantitative Information*. Englisch. 7th ed. Cheshire, Conn.: Graphics Press, 1986, 197 Seiten (cit. on p. 14).

Online Resources

[42] V. Agafonkin. *Leaflet — An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps*. URL: https://leafletjs.com/ (visited on 21/09/2018) (cit. on p. 44).

[43] H. Arthur. *clusterfck — JavaScript Agglomerate Hierarchical Clustering*. URL: https://harthur.github.io/clusterfck/ (visited on 21/09/2018) (cit. on p. 45).

[44] M. Bostock. *D3.js — Data-Driven Documents*. URL: https://d3js.org/ (visited on 21/09/2018) (cit. on p. 43).

[45] R. Cordell. *Oceanic Exchanges — Tracing Global Information Networks in Historical Newspaper Repositories, 1840–1914*. URL: http://oceanicexchanges.org/ (visited on 21/09/2018) (cit. on p. 9).

[46] J. Davies. *d3-cloud*. URL: https://github.com/jasondavies/d3-cloud (visited on 21/09/2018) (cit. on p. 45).

[47] Department of Computer Science and Technology, University of Cambridge. *B-Splines*. URL: https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node4.html (visited on 21/09/2018) (cit. on p. 12).

[48] Google LLC. *Google Cloud Translation API*. URL: https://translation.googleapis.com/translate (visited on 21/09/2018) (cit. on pp. 35, 48).

[49] Google LLC. *Google Geocoding API*. URL: https://maps.googleapis.com/maps/api/geocode/json (visited on 21/09/2018) (cit. on pp. 45, 48).

[50] Mapbox. *Mapbox API Documentation*. URL: https://www.mapbox.com/api-documentation/#maps (visited on 21/09/2018) (cit. on p. 44).

[51] A. Ronacher. *Flask — Web Development, One Drop at a Time*. URL: http://flask.pocoo.org/ (visited on 21/09/2018) (cit. on p. 45).

[52] A. Ronacher. *Jinja2*. URL: http://jinja.pocoo.org/ (visited on 21/09/2018) (cit. on p. 45).

[53] Square Inc. *Crossfilter — Fast Multidimensional Filtering for Coordinated Views*. URL: https://square.github.io/crossfilter/ (visited on 21/09/2018) (cit. on p. 45).

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature