Visualizing Large Collections of URLs Using the Hilbert Curve

Poornima Belavadi[1][0000−0003−1602−8369], Johannes Nakayama[1][0000−0001−9977−6471], and André Calero Valdez[2][0000−0002−6214−1461]

[1] Human-Computer Interaction Center, RWTH Aachen University, Campus Boulevard 57, 52076 Aachen, Germany {belavadi,nakayama}@comm.rwth-aachen.de
[2] Institute for Multimedia and Interactive Systems, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany
calerovaldez@imis.uni-luebeck.de

Abstract. Search engines like Google provide an aggregation mechanism for the web and constitute the main access point to the Internet for a large part of the population. For this reason, biases and personalization schemes of search results may have huge societal implications that require scientific inquiry and monitoring. This work is dedicated to visualizing data such inquiry produces as well as understanding changes and development over time in such data. We argue that the aforementioned data structure is very akin to text corpora, but possesses some distinct characteristics that requires novel visualization methods. The key differences between URLs and other textual data are their lack of internal cohesion, their relatively short lengths, and—most importantly—their semi-structured nature that is attributable to their standardized constituents (protocol, top-level domain, country domain, etc.). We present a novel technique to spatially represent such data while retaining comparability over time: A corpus of URLs in alphabetical order is evenly distributed onto the so-called Hilbert curve, a space-filling curve which can be used to map one-dimensional spaces into higher dimensions. Rank and other associated meta-data can then be mapped to other visualization primitives. We demonstrate the viability of this technique by applying it to a data set of Google search result lists. The data retains much of its spatial structure (i.e., the closeness between similar URLs) and the spatial stability of the Hilbert curve enables comparisons over time. To make our technique accessible, we provide an R-package compatible with the ggplot2-package.

Keywords: visualization techniques · text visualization · URL collections · computational social science

1 Introduction

In recent years, the world wide web has witnessed an exponential growth of the amount of data. In fact, the total volume of data has increased 30-fold worldwide since 2010 [17]. Generally, visualization methods for structured data are

Preprint. Final document published under DOI: 10.1007/978-3-031-14463-9_18 by Springer

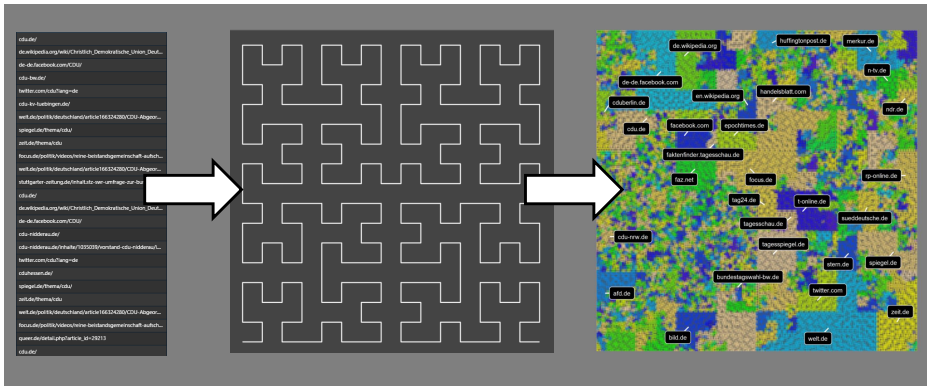

Fig. 1. We visualize large collections of URLs by ordering them alphabetically and taking advantage of the spatial stability of the Hilbert curve. The resulting visualizations provide a spatial mapping of domains and enable comparisons over time.

inherently more mature than for unstructured data and a relatively established repertoire of visualization techniques has been around for some time [21,34]. However, while it is even challenging to process the large amounts of newly generated structured data, it is the unstructured data that adds seemingly untamable complexity. Much of the newly generated data is unstructured textual data from Social Media sites, media outlets, blogs and forums, and many other places online. Visualization is a crucial instrument to reduce complexity, but methods for text visualization remain largely exploratory. As is characteristic for unstructured data, **textual data exhibits high levels of diversity and variability** which makes its visualization an inherently hard task.

Although textual data can be visualized directly (e.g., word clouds), a more common approach is the visualization of extracted structured meta-constructs like lengths, sentiments, or frequencies (e.g., TF-IDF, bag-of-words models, etc.). For many textual data types and adjacent data domains, the methods for visualization remain largely exploratory. We discovered one of these data domains in our **inquiries into the personalization of web search results** from Google searches. Web search engines like Google provide an access point for information retrieval on the Internet for a large number of users. Personalization of Google search results might thus have a large societal effect in terms of a biased representation of current events. The scientific inquiry into this problem area remains difficult because Google search result lists are not easy to come by on a large scale. In a crowd-sourced data set of Google search result lists, we found that the resulting data can be roughly structured in large collections of URLs that are associated with a user, a search keyword, a time, and a rank.

Our Contribution. This kind of data was inaccessible for us with the current state-of-the-art of data visualization because of its unstructured nature and the specific characteristics of URLs as a data type which we will explain further in this paper. **We present a technique to visualize such data** to make it

more readily accessible for scientific inquiry and monitoring. We further present an implementation of this procedure in the programming language R, which makes use of the implementation of the grammar of graphics *ggplot2* [37]. Moreover, we share all study materials and code on the Open Science Framework (link to the repository: https://osf.io/rnkyj).

2 Related Work

The problem that we formulate in this work does not fit easily into existing categories of related work. In the following sections, we review the literature on areas that are adjacent to the problem domain that we are addressing. Visualizing URLs falls into the domain of *text visualization* (subsection 2.1). More specifically, the data collections that we target for visualization can be thought of as *corpora* (subsection 2.2). However, URLs could also be seen as labels or categories. So it could also be seen as label visualization. Apart from these general considerations, we further review related work on *monitoring of online media streams* (subsection 2.3) as the technique we propose in this paper encompasses the aim of making data accessible for scientific inquiry and monitoring. Lastly, our approach falls into the broader scope of *map-like visualizations* which we briefly touch on in subsection 2.4.

2.1 Visualizing textual data

With the vast amounts of textual data that are available online today, proper visualization techniques are essential to reduce complexity. The literature in the field, especially when it comes to mature, sophisticated methods, is still surprisingly sparse. Kucher and Kerren [23] provide an ongoing visual survey of the field which, at the time of this work, contains 440 publications with **text visualization methods**. It is implemented as a web-based survey browser that lets users filter the works according to a **taxonomy** by the authors (available at: https://textvis.lnu.se/). The categories of this taxonomy are *analytic tasks*, *visualization tasks*, *data* (*data source* and *data properties*), *domain*, and *visualization* (*dimensionality*, *representation*, and *alignment*).

Using this tool, we identified work on text visualization that is similar to what we are aiming to achieve with the technique proposed in this paper. Following is the list of filters applied— *Analytic Tasks:* Event Analysis, Trend/-Pattern Analysis, *Visualization Tasks:* Clustering/Classification/Categorization, Overview, Monitoring, *Data Source:* Corpora, *Data Properties:* Time-series, *Domain:* Online Social Media, Other, *Visualization Dimensionality:* 2D, *Visualization Representation:* Pixel/Area/Matrix, *Visualization Alignment:* Other. We chose the *"Other"* option in places where the provided filters did not entirely match our requirements. At the time of this work, this query resulted in only three papers [30,8,2], shedding light on the rich scope for possible research in this area. However, **none of these matched our requirements**.

2.2 Visualization of Large Corpora

In computational linguistics and related fields, texts are often encountered within the context of *corpora*. Text corpora are large structured collections of texts that are rich in information and can be analyzed for a plethora of potential insights. Visualization often plays an important role in this pursuit. A common consideration in corpus visualization is the **distinction between the textual and inter-textual level** which need to be considered in conjunction [19,10,1]. One software solution for corpus analysis that includes both levels is the *DocuScope* software. It provides a set of visualization tools and summary statistics. DocuScope has been used for a wide variety of research purposes including the identification of factors that account for rhetorical variation in canned letters [19] and clarification of collaboration and authorship in the *Federalist Papers* [9]. Building on experiences with DocuScope, Correll, Witmore, and Gleicher [10] present new software for corpus visualization. Their application area is literary scholarship, or—more specifically—the exploration of corpora of tagged text. They introduce the tools CorpusSeparator and TextViewer which are intended to provide insights on the corpus and text level respectively. The software gives literary scholars the tools to quickly identify conspicuous sections of the data in overview and then to zoom into these parts for a more detailed look. **Overview is achieved through visualization of a principal component analysis (PCA)**.

While text corpora can be of a literary nature, the Internet has enabled the collection of large textual data sets in settings of computer-mediated communication (CMC). Abbasi and Chen [1] studied CMC archives and introduced **a technique to perform classification analyses on the textual level**. CMC archives contain rich information about social dynamics and they are often characterized by high numbers of authors, forums, and threads. At the same time, they are inherently hard to navigate. Abbasi and Chen introduce the *Ink Blot* technique which lends itself well to a multitude of visual classification tasks. The technique overlays text with colored ink blots which highlight specific patterns that characterize a particular text within the context of the corpus from which it was derived.

2.3 Monitoring of Online Media Streams

The visualization of corpora is traditionally aimed at analysis, but with the inexorable amount of data that is created constantly, the visualization of online media streams has recently sparked an increasing interest. It has resulted in the development of many media monitoring platforms and tools *(www.ecoresearch.net, www.noaa.gov)* [32] whose goal is to detect and analyze events and **reveal the different perceptions of the stakeholders and the flow of information**. Such tools often use information extraction algorithms to be able to work with large collections of documents that differ in formatting, style, authorship, and update frequency [32]. Some of the commonly used visualizations by these tools that

help in uncovering the complex and hidden relations within the document collection are map-like visualizations, tag clouds, radar charts, and keyword graphs.

Data streams found online can be a rich data source, but their visualization often requires ad-hoc solutions which means that they are cost-intensive. It is thus all the more important to abstract from patterns that occur in online data streams and introduce techniques that apply to a certain problem across contexts. One example is the visualization of news streams, for which Cui et al. [11] introduce a visualization scheme called *TextWheel*. They make use of familiar visual metaphors (Ferris wheel, conveyor belt) to display the development of news streams in one coherent and comprehensible scheme. The technique enables **a comprehensible display of temporal developments** and the authors demonstrate the technique on two example data sets.

2.4 Map-Like Visualization

In visualization research, the term *map-like* can be found describing visualizations that combine the features of cartographic maps to represent abstract data [16]. This representation takes advantage of our ability to recall spatial information or interpret spatial relations between the elements in a map as a similarity measure. Evidence for the **cognitive benefits of using maps** abounds [13]. It is not surprising to see that one of the oldest forms of visualizing spatial data has been in the form of *cartographic maps*.

Investigating ways of leveraging the benefits of maps in visualizing data has been the research focus of the field. In their work on reviewing the state-of-the-art in map-like visualizations, Hograefer et al. [16] have classified the visualizations based on the availability of geographical context into two groups—*schematization* and *imitation*.

Schematization refers to visualizations that are "map-like", where the cartographic **maps are transformed into abstract visualizations** showing emphasis on the thematic data that is spread over a geographical frame of reference. Applying schematization improves the readability by simplifying the map and maintaining the geographic topology, which aids the users to orient themselves in the data space. The visualizations in this technique involve a fundamental trade-off of emphasizing between the visualization of data by applying more schematization and keeping the geographical topology recognizable [4].

Imitation is opposite to schematization, here the abstract visualizations are made to look "map-like" by refining the complex and irregular areas and lines. Imitation depicts **abstract, non-spatial data as visual primitives in a two-dimensional display** by assigning it a position on the plane [16]. To be considered "map-like", however, the positions should achieve a meaningful measure of the distance between visual primitives. Meaningful proximity can be achieved by mapping the dimensions of the data onto the visualization axes, which also helps in understanding the similarity between the data. **Dimensionality reduction methods** like multi-dimensional scaling (MDS), principal component analysis (PCA), and t-distributed stochastic neighbor embedding (t-SNE) **are used** to map n-dimensional data and their distances to 2D [35].

Mapping large data sets into a space suitable for map-like visualizations is both conceptually and computationally hard and requires specialized techniques. Keim [20] proposes the use of space-filling pattern schemes in a new paradigm for visualization that he terms *pixel-oriented techniques*. These techniques use every pixel of a plot panel to maximize the amount of data that can be visualized, making them particularly suitable for the visualization of big data sets. To utilize every pixel space-filling pattern schemes or space-filling curves are used to map the data to a 2D position [18,25]. These curves have the advantage that **they retain clusters that are present in one dimension** and make them easily discernible in two dimensions. Space-filling curves fall under the *Imitation* technique of visualizing data and are mostly used in combination with a grid. When all the cells that are surrounded by a curve are joined to form a border, the resulting outline forms an area on the "map" [33].

A detailed look into the related literature reinforced the impression that the visualization technique we develop in this paper is novel for visualizing large corpora of URLs and sheds light on the vast possibilities available for research. So far, previous work done in visualizing textual data does not apply to visualizing long lists of URLs because of the structural differences between URLs and normal text data. With the visualization technique applied in this paper, we focus on improving the understanding of personalization. Given the ubiquitous nature of personalization algorithms, this presents a current and critical challenge.

3 Method

Almost a decade ago, Eli Pariser coined the term *filter bubble* [28] as a metaphor for the personalization of web content and the problems that arise because of it. Even though this topic has garnered a lot of public attention since then, data which could be used to address the questions around this phenomenon is still surprisingly sparse. In the following paragraphs, we outline how we addressed the problem of getting an **overview over web search engine personalization** through visualization. Firstly, we describe the requirements of a data set that could be used for such a task.

3.1 Data: Requirements and Description

Procuring the data necessary to address the issues around personalization on web search engines is a challenging task. Krafft, Gamer, and Zweig [22] collected an appropriate data set for the task at hand through "data donation". "Data donors" were asked to install a browser plugin that conducted Google searches in regular time intervals for specific search terms over an extended period of time. In their case, they used the major German political parties and the names of the primary candidates of each party as search terms and **collected Google search results in the months leading up to the German federal election 2017**. The resulting data set is a collection of search engine result lists. Each

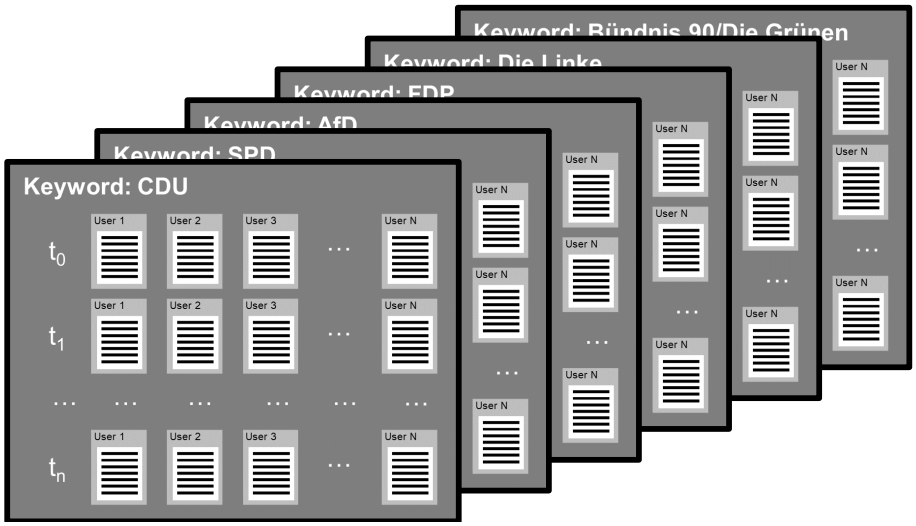

Fig. 2. Structure of a compilation of Google search result lists from different users over an extended period of time.

data item is a list of URLs which is associated with a user, a timestamp, and a search keyword (Figure 2).

Importantly, the result pages that are collected have to be considered as *ranked* lists and not merely as sets. There are strong indications that the rank of a search engine result is strongly related to the attention users pay to the result [24]. While metrics to compare ranked lists exist [36], we advocate for breaking the structure of the data down to make potential results more accessible for interpretation. **We treat the data set as a collection of URLs** each of which is associated with a user, a timestamp, a search keyword, and a rank. The atomic unit of this consideration is thus a singular URL instead of a ranked list of URLs which makes it considerably easier to find a spatial mapping for visualization.

3.2 Problem

URLs are a very particular type of data with specific characteristics that need to be carefully considered. URLs were preceded by a standard access format to documents on the world wide web called Uniform Resource Identifiers (URIs) which followed the same paradigm: object addresses as strings **with a standardized syntax** [5].

URLs were adapted as the standard addressing scheme of web content and with heavy growth rates early on, users were in dire need of aggregation measures and search capabilities. One of the first web search engines was provided by the so-called World Wide Web Worm [26] which cataloged web resources hierarchically and provided keyword search capabilities. Strikingly, it followed a

paradigm for web searches that would later become almost universal: curating web search results as a "bulletin board", a list of entries that are relevant to the search specifications, with a hyperlink to the location specified by the associated URL.

While this system provided some much needed complexity reduction at its time, **search engines** were soon confronted with serious scalability issues. These issues were addressed by Sergey Brin and Larry Page who introduced their search engine Google [6] which became a staggering success and the entry point to the world wide web of an unfathomably large number of users. Nowadays, search engines—and first and foremost Google—present the **primary access point to the world wide web** for a large number of users. This means that large portions of a population could get their information on current topics from web searches. For instance, in the 3rd quarter of 2020, the search term "news" is the fifth most popular search term on Google [12]. This presents a potential societal challenge of the following kind: A suspicion concerning search engine results is that they **arrange content differently for different users**, leading to biases and distortions in the users' perceptions. The question is how severe of a problem this actually is and whether the distortion varies over time and between search terms.

3.3 Why not Map into a Metric Space?

Intuitively, one might think that a viable solution is mapping the data into a (high-dimensional) metric space and to then apply methods like principal component analysis or multi-dimensional scaling for mapping into the 2D plane. However, there are several problems with this solution that, in our view, disqualify it for the problem at hand. First of all, **feature extraction on URLs is not viable** because URLs are short and lack the semantic information content that textual data usually possesses. Second, defining a **string metric** on the set of URLs and treat it as a metric space does not work either. Many **distance measures** that are applicable to strings do not meet the requirement for a metric space in the first place or are otherwise not viable. For instance, the Hamming distance is only defined on strings of the same length and the Cosine distance does not satisfy the triangle inequality. Levenshtein, Damerau-Levenshtein, and Jaccard distance satisfy the triangle inequality, but do not produce sensible results for the problem at hand because of the standardized, semi-structured nature of URLs. String comparison techniques like the aforementioned compare the characters of the whole string. However, character distances in URLs do not equate to meaningful distances in websites found under these URLs (e.g., compare google.com and moodle.com).

3.4 Using Space-Filling Curves

Space-filling curves are **continuous and bijective mappings from one-dimensional into higher-dimensional space**. As the name suggests, the curve passes through every point of an n-dimensional space, which is achieved

by folding a one-dimensional line an infinite number of times [3]. Space-filling curves were first discovered by Giuseppe Peano in 1890 [29] and several algorithms mapping one-dimensional lines into higher-dimensional spaces have been discovered since. We will demonstrate that much of the spatial structure in a collection of URLs with the aforementioned meta-data can be retained by ordering the URLs alphabetically and subsequently mapping them into 2D space by means of the Hilbert curve, a space-filling curve that was proposed by the mathematician David Hilbert shortly after Peano's original discovery [15]. The Hilbert curve is a pattern that recursively splits a unit hypercube into quadrants and folds a line into those quadrants according to a set of folding and rotation rules. The simplest and most accessible type of the Hilbert curve is its 2D-instantiation. The first four iterations of this recursive pattern are displayed in Figure 3. Following, we refer to the 2D-instantiation when we use the term Hilbert curve.

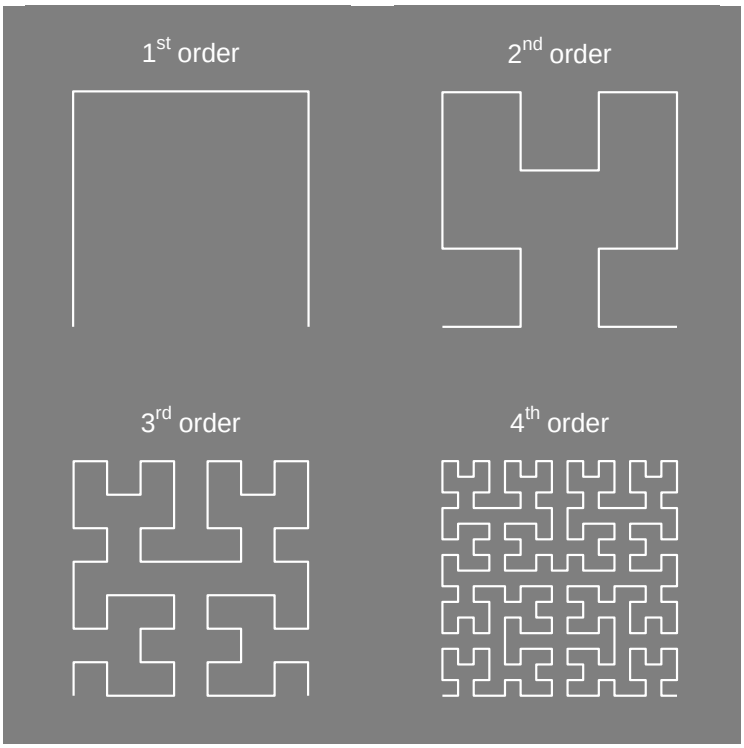

Fig. 3. First to fourth order Hilbert curves.

Hilbert's space-filling curve has fascinated mathematicians for a long time, mostly—however—for its aesthetic appeal. **Useful application areas** have only been proposed relatively recently. Primarily, the Hilbert curve seems to lend

itself well to overview and monitoring tasks. Areas in which the Hilbert curve has been applied include the visualization of genomic data [3] and the monitoring of network traffic [31].

Using the Hilbert curve for visualization comes with the benefit that **points that are close in 1D space are also close in the 2D visualization**. This is not necessarily the case in the other direction, where at folding points (e.g., at the centre of the plot), it is possible that data items from distant positions in a vector are close to each other on the 2D plane. However, this effect is weaker for the Hilbert curve than for other space-filling curves [3].

Another benefit is the curve's stability when increasing the resolution of the input data. Increasing the number of iterations yields a longer curve which is more densely folded into the same space, ensuring that **points will always map to a similar position**. This even applies when a small amount of non-uniformly distributed data is introduced to the data.

URLs possess an internal structure that is conducive to a hierarchical treatment of position within a string—which is characteristic for alphabetical orderings. Figure 4 shows a typical URL along with an appraisal of the relative importance of each of its constituents. We suggest that protocol (mostly "https") and the "www." specification do not contain useful information when it comes to evaluating the exposure of users to different web content which is why we excluded these parts in pre-processing. We would further argue that the remaining part of the URL is structured somewhat hierarchically with regard to the importance of informational content. The first elements of the remaining URL are the domain, which specifies the host, and the top-level domain, which can give insight into the location of that host (if it is a country domain) or the kind of institution it represents (".org", ".edu", etc.). Lastly, there is the path to the resource which is usually hierarchically structured from abstract to concrete.

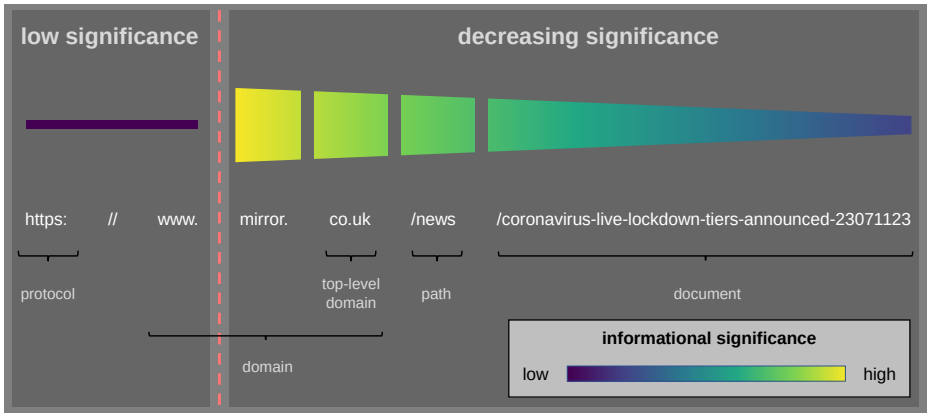

Fig. 4. URL structure and significance estimation.

We propose that for the reasons outlined above, **mapping URLs onto the Hilbert curve in alphabetical order preserves the clustering** essential to the task of visualizing personalization of search engine results. In the following sections, we document an algorithm to achieve a visualization of this kind and present an R package that enables practitioners to use this technique in their own inquiries into the problem domain.

3.5 Implementation

To create a Hilbert curve visualization it is first necessary to understand that n^{th} order Hilbert curves have a number of corners that powers of four. The first order has 4 corners, the second order hast $4^2 = 16$ corners, the third order has $4^3 = 64$ corners, and so on (see Figure 3). For a map-like visualization that projects individual list entries to a 2D position, we want to **map the entries onto the corners of a Hilbert curve**. It is also possible to map the entries onto the lines connecting the corners, however this would remove the benefit of mapping all entries into the smallest possible 2D representation by creating empty space.

Given the number of corners in a Hilbert curve, it is easiest to also visualize lists of items that have a length that are powers of four. This cannot generally be assumed. To circumvent this problem, we scale the rank numbers of the entries to the next higher power of 4. If we have 10 entries, we use 16 corners. Entry number 8 would be mapped to $\text{ceil}(8/10 \times 16) = 13$. Each entry is mapped to another unique corner number, creating skipped corners here and there. This creates a **reduced Hilbert curve** (see Figure 5) which nevertheless retains all of the important properties mentioned above.

From the list of corner numbers we generate x and y coordinates, using an **iterative Hilbert function**. This function "performs the mappings in both directions, using iteration and bit operations rather than recursion. It assumes a square divided into n by n cells, for n a power of 4, with integer coordinates, with $(0, 0)$ in the lower left corner, $(n - 1, n - 1)$ in the upper right corner, and a distance d that starts at 0 in the lower left corner and goes to $n^2 - 1$ in the lower-right corner"[38].

The benefit of creating a visualization using this approach is that it retains relative stability of positions for large data sets (Figure 6). When individual data points are added into the data, the displacement of existing data tends towards zero for large data sets (Figure 7). To demonstrate this, we simulated 10 runs of iteratively adding 2200 points of random data into a Hilbert curve. All runs show very little variation regarding root mean squared displacement.

3.6 R package

To enable reproducibility, we implemented an R package that provides a function to produce a Hilbert visualization for strings. The package is hosted on GitHub

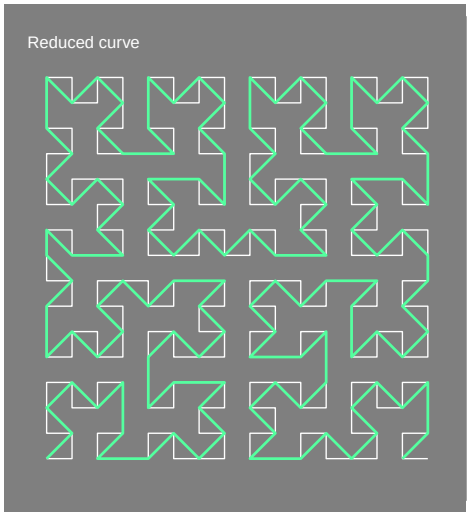

Fig. 5. By scaling the ranks of position we can map an arbitrary number of entries to spatially stable positions. The line in green shows the new reduced curve.

(https://github.com/Sumidu/gghilbertstrings). The conversion of ranks into positions, which is the computationally expensive part of the function, was implemented in C++ and made interoperable with Rcpp [14]. The `gghilbertstrings` package enables the creation of graphics like the ones we show in this paper. The core function of the package returns a `ggplot2` object [37]. The major advantage of building on the existing visualization framework provided by `ggplot2` is the flexibility that it grants with regard to customization.

4 Demonstration

For demonstration, we use a **data set collected during the lead-up of the German federal election 2017** [22]. It consists of the Google search results for 16 search terms of over 4000 users over a period of 86 days. The search results were collected daily by a browser plugin that "data donors" could install to contribute to the research project. The keywords were the names of the major German political parties and their top candidates.

Before visualization, the data is pre-processed as described above: First, we remove the low information section in the beginning of the URL, then we collect them in one large list with associated meta-data and sort them alphabetically. For this demonstration, we opted for subsetting the data and only use the result lists for the search keywords "AfD" (right-wing populist party) and "CDU" (christian conservative party). Figure 8 shows the spatial mapping of all URLs that occurred in the respective result lists over the entire time frame covered by the data. **Each colored region in this plot indicates a domain** and the 30 domains that occurred most frequently are indicated with labels pointing to the

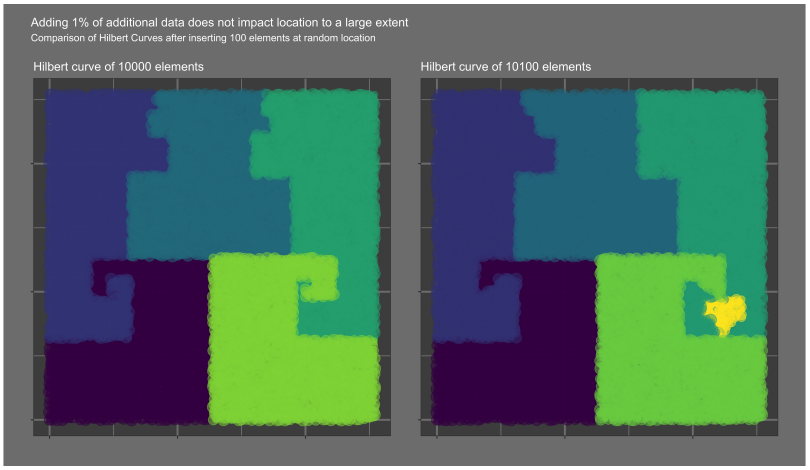

Fig. 6. By adding 1% of additional data (shown in yellow) to an existing Hilbert curve at a random location, we can see that most areas are spatially stable.

mean location of URLs in the respective region. Note that larger regions indicate domains that occurred more frequently.

Building on the familiar visual metaphor of topological maps, we chose a color space that is typically used in that domain to encode the regions. As there are 4074 different domains in the data set, we divided the color scale accordingly and randomly assigned each domain one of the generated colors. **The resulting map-like visualization will serve as a reference point** to enable comparisons of different patterns over time.

The landscape for a single point in time can then be displayed by stratifying the data by time. The actual search results for a particular day are mapped onto the reference map as round white markers with high transparency values (akin to viewing clouds from a birds-eye view) to account for overplotting. The size of the markers is contingent of the rank of that particular URL in the search result list that it stems from. Corresponding to their higher relevance, larger markers thus indicate higher-ranked results.

Figure 9 displays the spatial representations of the search result lists of all users for the terms "AfD" and "CDU" on two different days, one of which (September 24, 2017) was the day of the German federal election. The middle-left section of the reference map shows many small domain regions. These account for the websites of local organizations of the CDU party. The **variability in the search results** displayed in the bottom-row plots (CDU) are thus **accounted for by local personalization**. It can be seen that this local personalization is weaker on the day of the election, which is likely due to more media coverage of the CDU on that day, indicated by a higher concentration of solid-looking markers in the bottom right corner of the plot where the regions for *spiegel.de*, *welt.de*, and *zeit.de* (three of the biggest German media outlets) are located. In

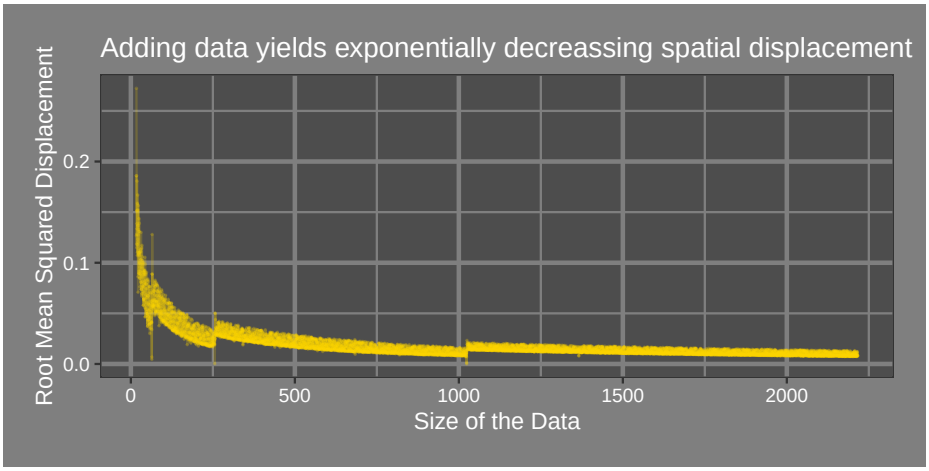

Fig. 7. By consecutively adding data to a Hilbert curve, we see that the amount of displacement continuously shrinks. Occasionally when the order of the curve increases, displacement shortly spikes.

comparison, there is relatively low variability in the search results for the term "AfD". A possible explanation for this might be that the party has only existed for a few years and there are not as many local organizations which have websites, resulting in less local personalization of search results. Still, upon closer inspection, one can identify more accentuated clusters of URLs in the regions representing larger media outlets on the day of the election, indicating a higher media coverage than on a reference day two months before the election.

For long-term monitoring and identification of conspicuous patterns in the data, **these graphics can further be animated** with regard to time. Animations of this kind can then be searched for anomalies which in turn could spark further inquiry into that region of the data.

5 Discussion

Text visualization has experienced a surge in research interest since about 2007 [23]. Still, there is a lot of scope for research in this area, particularly because of the wide variety of structural complexities that different types of textual data come with. One such data type that requires dedicated attention is presented by URLs and collections thereof. **URLs differ from conventional text** data in many regards, e.g., in that they cannot be tokenized and often occur in large non-cohesive collections.

We argue that the paradigm of using space-filling curves is promising with regard to creating coherent visualizations of such data. Castro and Burns [7] found that the Hilbert curve method produces an optimal mapping where an arbitrary block of information will be divided a minimum possible number of times in the

mapped space [7]. This finding was based on the result of an analytical approach by Mokbel, Aref, and Kamel [27]. With regard to URLs, we showed that an **alphabetical ordering retains the meaningful similarities** between different URLs (same domain) while still allowing for a spatial mapping that makes different regions distinguishable. Even though the closeness of adjacent regions is not actually indicative of closeness between the domains, the resulting visualization grants a high-level overview of the data that is interpretable and easy to create.

When streaming data is introduced into the visualization, long-term stability over time cannot be guaranteed. If the distribution of domain names in the newly introduced data stays stable, so does the visualization. If this is not the case—e.g., if a large number of URLs with the same domain name is introduced—the positions in the visualization are shifted. However, this would happen in a homogeneous fashion. One idea to address this problem would be to introduce "white noise" into the data: large amounts of equally distributed data points that will not be visualized. Even though this would increase the computational cost as well as **introduce randomness into the visualization**, this adjustment would **improve stability over time**. We also observed that if the input data set contained a large number of links with an identical domain, the resulting visualization would become distorted. In this case, adding noise to the data might not be suitable because the amount of noise would have to be significantly higher than the amount of data. We focus on improving the algorithm to overcome this problem in our future work.

Corpora of URLs might be encountered in different contexts and the technique that we present here may be applicable across different areas of inquiry. However, several caveats have to be taken into consideration. For instance, the naive accumulation of URLs is not necessarily viable in the following cases:

- When link shorteners are used, the lexicographical ordering of URLs loses any meaning.
- When websites do not utilize the URL paths in a meaningful fashion, the only meaningful unit of observation left is the domain.

Still, the technique is generally applicable to text where the lexicographical ordering of entries carries most of the meaning. Other contexts where one might encounter such data include media analyses of hyperlinks used in various news outlets, social media, or any other large scale web-resources.

In general, the application domain lies primarily in the computational social sciences. We thus opted for implementing our approach in the R language, which is a popular choice in this field of research. The package API is interoperable with `ggplot2` [37], an implementation of the grammar of graphics [39] which provides future users with the capability to easily customize the generated plots. Our implementation enables the production of visualizations at small computational cost, making it a viable choice in both interactive analysis and real-time monitoring.

6 Conclusion

In this paper, we addressed the complex issue of visualizing multiple long lists of URLs. We extract meaningful structural information from one-dimensional data by using alphabetical ordering as a proxy for the similarity between URLs and use their ranks to map them into 2D space using the Hilbert curve technique.

We demonstrated our technique on a data set containing Google search results of over 4000 users. Practitioners can replicate this method with the R package that we provide on GitHub. In future inquiries, we will address the issue of stability over time for streaming and distorted data.

Acknowledgements

We would like to thank Nils Plettenberg for his help in developing the the initial ideas of this project. This research was supported by the Digital Society research program funded by the Ministry of Culture and Science of the German State of North Rhine-Westphalia. We would further like to thank the authors of the packages we have used.

References

1. Abbasi, A., Chen, H.: Categorization and analysis of text in computer mediated communication archives using visualization. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. pp. 11–18. JCDL '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1255175.1255178, https://doi.org/10.1145/1255175.1255178
2. Almutairi, B.A.A.: Visualizing patterns of appraisal in texts and corpora. Text & Talk **33**(4-5), 691–723 (2013)
3. Anders, S.: Visualization of genomic data with the hilbert curve. Bioinformatics **25**(10), 1231–1235 (2009)
4. Barkowsky, T., Latecki, L.J., Richter, K.F.: Schematizing maps: Simplification of geographic shape by discrete curve evolution. In: Spatial Cognition II, pp. 41–53. Springer (2000)
5. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H., Secret, A.: The world-wide web. Communications of the ACM **37**(8), 76–82 (1994). https://doi.org/10.1145/179606.179671
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks **30**, 107–117 (1998), http://www-db.stanford.edu/~backrub/google.html
7. Castro, J., Burns, S.: Online data visualization of multidimensional databases using the hilbert space–filling curve. In: Visual Information Expert Workshop. pp. 92–109. Springer (2006)
8. Chi, E.H., Hong, L., Heiser, J., Card, S.K.: Scentindex: Conceptually reorganizing subject indexes for reading. In: 2006 IEEE Symposium On Visual Analytics Science And Technology. pp. 159–166. IEEE (2006)

9. Collins, J., Kaufer, D., Vlachos, P., Butler, B., Ishizaki, S.: Detecting collaborations in text comparing the authors' rhetorical language choices in the federalist papers. Computers and the Humanities **38**(1), 15–36 (2004)
10. Correll, M., Witmore, M., Gleicher, M.: Exploring collections of tagged text for literary scholarship. Computer Graphics Forum **30**(3), 731–740 (2011). https://doi.org/https://doi.org/10.1111/j.1467-8659.2011.01922.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01922.x
11. Cui, W., Qu, H., Zhou, H., Zhang, W., Skiena, S.: Watch the story unfold with textwheel: Visualization of large-scale news streams. ACM Trans. Intell. Syst. Technol. **3**(2) (Feb 2012). https://doi.org/10.1145/2089094.2089096, https://doi.org/10.1145/2089094.2089096
12. DataReportal, We Are Social, Hootsuite: Top google search queries worldwide during 3rd quarter 2020 (index value) [graph] (October 2020), https://www.statista.com/statistics/265825/number-of-searches-worldwide/, retrieved: November 30, 2020
13. DeLoache, J.S.: Becoming symbol-minded. Trends in cognitive sciences **8**(2), 66–70 (2004)
14. Eddelbuettel, D., François, R.: Rcpp: Seamless R and C++ integration. Journal of Statistical Software **40**(8), 1–18 (2011). https://doi.org/10.18637/jss.v040.i08, http://www.jstatsoft.org/v40/i08/
15. Hilbert, D.: über die stetige abbildung einer linie auf ein flächenstück. Mathematische Annalen **38**, 459–460 (1891)
16. Hogräfer, M., Heitzler, M., Schulz, H.J.: The state of the art in map-like visualization. In: Computer Graphics Forum. vol. 39, pp. 647–674. Wiley Online Library (2020)
17. IDC, Statista: Volume of data/information worldwide from 2010 to 2024 (in zettabytes) [graph] (May 2020), https://www.statista.com/statistics/871513/worldwide-data-created/, retrieved: November 19, 2020
18. Irwin, B., Pilkington, N.: High level internet scale traffic visualization using hilbert curve mapping. In: VizSEC 2007, pp. 147–158. Springer (2008)
19. Kaufer, D., Ishizaki, S.: A corpus study of canned letters: Mining the latent rhetorical proficiencies marketed to writers-in-a-hurry and non-writers. IEEE Transactions on Professional Communication **49**(3), 254–266 (2006). https://doi.org/10.1109/TPC.2006.880743
20. Keim, D.A.: Pixel-oriented visualization techniques for exploring very large data bases. Journal of Computational and Graphical Statistics **5**(1), 58–77 (1996)
21. Keim, D.A.: Information visualization and visual data mining. IEEE transactions on Visualization and Computer Graphics **8**(1), 1–8 (2002)
22. Krafft, T.D., Gamer, M., Zweig, K.A.: What did you see? a study to measure personalization in google's search engine. EPJ Data Science **8**(1), 38 (2019)
23. Kucher, K., Kerren, A.: Text visualization techniques: Taxonomy, visual survey, and community insights. In: 2015 IEEE Pacific Visualization Symposium (PacificVis). pp. 117–121. IEEE (2015)
24. Lorigo, L., Haridasan, M., Brynjarsdóttir, H., Xia, L., Joachims, T., Gay, G., Granka, L., Pellacini, F., Pan, B.: Eye tracking and online search: Lessons learned and challenges ahead. Journal of the American Society for Information Science and Technology **59**(7), 1041–1052. https://doi.org/https://doi.org/10.1002/asi.20794, https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20794
25. Markowsky, L., Markowsky, G.: Scanning for vulnerable devices in the internet of things. In: 2015 IEEE 8th International conference on intelligent data acquisition

and advanced computing systems: technology and applications (IDAACS). vol. 1, pp. 463–467. IEEE (2015)
26. McBryan, O.A.: Genvl and wwww: Tools for taming the web. In: In Proceedings of the First International World Wide Web Conference. pp. 79–90 (1994)
27. Mokbel, M.F., Aref, W.G., Kamel, I.: Performance of multi-dimensional space-filling curves. In: Proceedings of the 10th ACM international symposium on Advances in geographic information systems. pp. 149–154 (2002)
28. Pariser, E.: The filter bubble: What the Internet is hiding from you. Penguin UK (2011)
29. Peano, G.: Sur une courbe, qui remplit toute une aire plane. Mathematische Annalen **36**(1), 157–160 (1890)
30. Rohrer, R.M., Ebert, D.S., Sibert, J.L.: The shape of shakespeare: visualizing text using implicit surfaces. In: Proceedings IEEE Symposium on Information Visualization (Cat. No. 98TB100258). pp. 121–129. IEEE (1998)
31. Samak, T., Ghanem, S., Ismail, M.A.: On the efficiency of using space-filling curves in network traffic representation. In: IEEE INFOCOM Workshops 2008. pp. 1–6. IEEE (2008)
32. Scharl, A., Hubmann-Haidvogel, A., Weichselbraun, A., Wohlgenannt, G., Lang, H.P., Sabou, M.: Extraction and interactive exploration of knowledge from aggregated news and social media content. In: Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems. pp. 163–168 (2012)
33. Schulz, C., Nocaj, A., Goertler, J., Deussen, O., Brandes, U., Weiskopf, D.: Probabilistic graph layout for uncertain network visualization. IEEE transactions on visualization and computer graphics **23**(1), 531–540 (2016)
34. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE symposium on visual languages. pp. 336–343. IEEE (1996)
35. Skupin, A., Fabrikant, S.I.: Spatialization methods: a cartographic research agenda for non-geographic information visualization. Cartography and Geographic Information Science **30**(2), 99–119 (2003)
36. Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. ACM Transactions on Information Systems (TOIS) **28**(4), 1–38 (2010)
37. Wickham, H.: ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York (2016), https://ggplot2.tidyverse.org
38. Wikipedia contributors: Hilbert curve — Wikipedia, the free encyclopedia (2020), https://en.wikipedia.org/w/index.php?title=Hilbert_curve&oldid=990914971, [Online; accessed 3-December-2020]
39. Wilkinson, L.: The grammar of graphics. In: Handbook of Computational Statistics, pp. 375–414. Springer (2012)

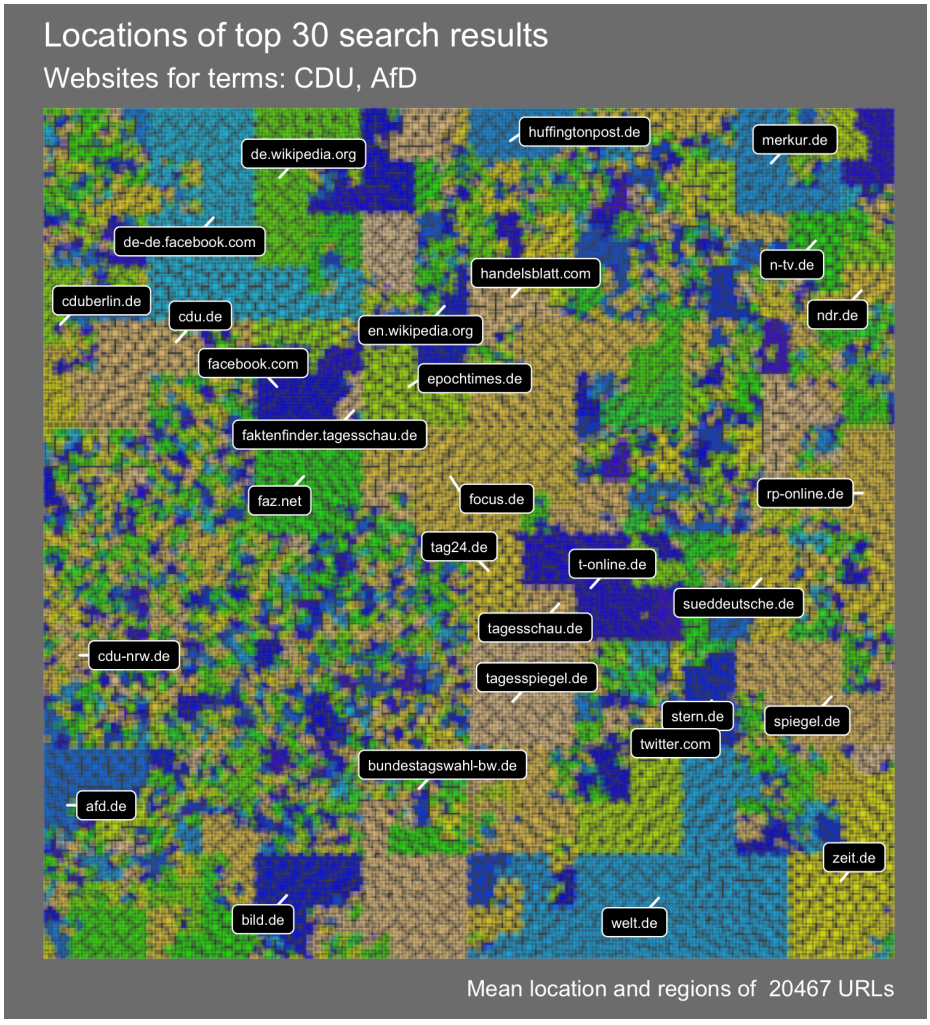

Fig. 8. Relative locations of the top 30 domains for the search terms AfD and CDU. Regions are colored by domain. Larger areas reflect more different URLs returned for the same domain.

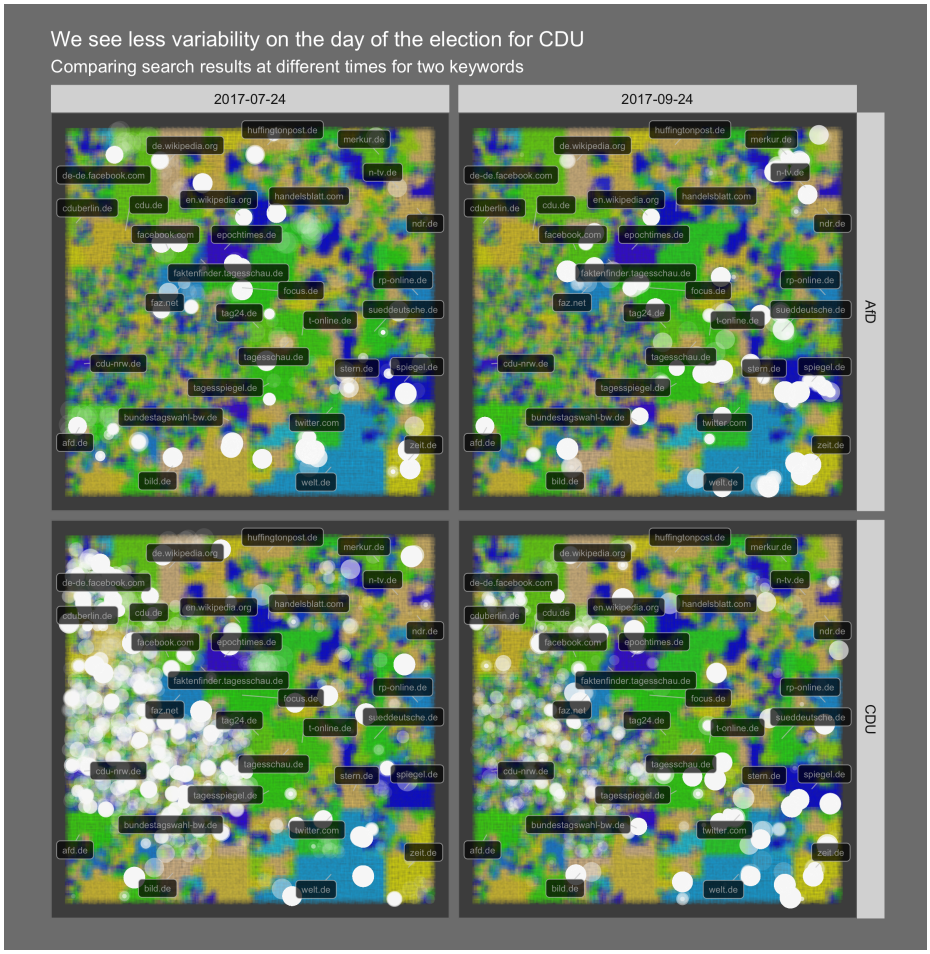

Fig. 9. Comparison of search results of two days. Election day (right column) shows a less diverse set of results for the leading party CDU (bottom row) than two months before the election. Some results stay the same over time. Several of these can we viewed as an animation to see changes over time.