SolarView

Citation for published version (APA):

Castermans, T., Verbeek, K., Speckmann, B., Westenberg, M. A., Koopman, R., Wang, S., van den Berg, H., & Betti, A. (2019). SolarView: low distortion radial embeddings with a focus. *IEEE Transactions on Visualization and Computer Graphics*, 25(10), 2969-2982. <https://doi.org/10.1109/TVCG.2018.2865361>

DOI:

[10.1109/TVCG.2018.2865361](https://doi.org/10.1109/TVCG.2018.2865361)

Document status and date:

Published: 01/10/2019

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

SolarView: Low Distortion Radial Embedding with a Focus

Thom Castermans, Kevin Verbeek, Bettina Speckmann, Michel A. Westenberg,
Rob Koopman, Shenghui Wang, Hein van den Berg, and Arianna Betti

Abstract—We propose a novel type of low distortion radial embedding which focuses on one specific entity and its closest neighbors. Our embedding preserves near-exact distances to the *focus entity* and aims to minimize distortion between the other entities. We present an interactive exploration tool *SolarView* which places the focus entity at the center of a “solar system” and embeds its neighbors guided by concentric circles. *SolarView* provides an implementation of our novel embedding and several state-of-the-art dimensionality reduction and embedding techniques, which we adapted to our setting in various ways. We experimentally evaluated our embedding and compared it to these state-of-the-art techniques. The results show that our embedding competes with these techniques and achieves low distortion in practice. Our method performs particularly well when the visualization, and hence the embedding, adheres to the solar system design principle of our application. Nonetheless—as with all dimensionality reduction techniques—the distortion may be high. We leverage interaction techniques to give clear visual cues that allow users to accurately judge distortion. We illustrate the use of *SolarView* by exploring the high-dimensional metric space of bibliographic entity similarities.

Index Terms—Dimensionality reduction, radial embedding, visualizing distortion.

1 INTRODUCTION

A MAJOR challenge in information visualization is the accurate visualization of multivariate data. Many techniques have been developed over the years for different types of data from various fields. Sometimes it is possible to leverage specific properties of the data, such as having numerical values, to create parallel coordinates [1] and scatterplot matrices [2]. Multiple linked views are also often employed [3], as are small multiples [4]–[7]. When the data points include coordinates, every entity in the data set can be represented by a position in a multi-dimensional space. One can then consider the distances in that space to see which data points are close to each other—which can be interpreted as those data points representing similar entities. However, when the number of coordinates is high and the data points thus lie in a high-dimensional space, it is harder to understand distances between points and to accurately visualize entities and their mutual similarities. Therefore techniques are often employed to reduce the dimensionality of a data set—most often to 2D—in such a way that distances between points are preserved as well as possible. Standard examples of such dimensionality reduction techniques are multidimensional scaling and principal component analysis. Distance is a very strong visual cue [8] and can in this setting be used effectively to communicate the similarity of entities.

It is generally impossible to reduce the dimension while maintaining all distances between data points. The *distortion*

is the ratio between distances in the original space and distances in the lower dimensional space (we give a formal definition in Section 4). The fact that there is always (significant) distortion is often ignored in visualization techniques that use dimensionality reduction. This means that users, when they have developed trust in visualizations that do not indicate distortion, may draw incorrect conclusions from these visualizations [9]–[11].

1.1 Our contribution

We present a new technique for dimensionality reduction into two dimensions. Our input consists of a set of data points (entities) in a high-dimensional metric space, one of which is a *focus entity*. We describe a novel low distortion radial embedding which preserves near-exact distances to the focus entity and aims to minimize distortion between the *neighbor entities*. None of the standard dimensionality reduction techniques employed in information visualization are designed to control specific distances in the embedding.

Neighbor entities who have a similar distance to the focus entity pose a challenge for each embedding algorithm; they are restricted to lie essentially on the same circle around the focus entity and their interactions have a strong influence on the distortion of the embedding. Our algorithm handles such entities explicitly by grouping all neighbor entities by their distance to the focus entity into a user-defined set of buckets. Each bucket is represented by a circle around the focus entity, where the radius of the circle corresponds to the center of the bucket distance range. We first compute a low-distortion metric embedding of the entities in each bucket onto the corresponding circle. We then rotate the circles and iteratively improve the entity positions to minimize inter-bucket distortion. At the end of this process we have found the angular position for each

- T. Castermans, K. Verbeek, B. Speckmann and M.A. Westenberg are with the TU Eindhoven.
[t.h.a.castermans/k.a.b.verbeek/b.speckmann/m.a.westenberg]@tue.nl
- R. Koopman and S. Wang are with OCLC.
[rob.koopman/shenghui.wang]@oclc.org
- A. Betti and H. van den Berg are with the University of Amsterdam.
[a.betti/h.vandenbergh]@uva.nl

Manuscript received MONTH DAY, YEAR; revised MONTH DAY, YEAR.

entity. We now have two options: (i) move each entity on the line through the focus within the distance range of its bucket onto the exact position realizing its distance to the focus, or (ii) snap entities to their bucket circles. The second option incurs a small distortion but we believe that it improves the overall clarity of the visualization. Our interactive tool *SolarView* leverages interactivity to offer a clear indication of the distortion of this dimensionality reduction.

Organization. We begin by surveying related work in Section 2. To place our technical contribution into context, we briefly analyze in Section 3 the requirements for the interactive exploration of high-dimensional data sets using focus entities. Here we discuss under which circumstances focus entities might be used and when a radial embedding might be desirable. In Section 4 we then explain our radial embedding algorithm. Section 5 describes a prototype implementation of *SolarView* that can be used to explore bibliographical records from OCLC’s WorldCat [12] data set. This section also discusses visualization details, such as smooth transitions to other focus entities.

In Section 6 we experimentally evaluate our embedding and compare it to state-of-the-art dimensionality reduction and embedding techniques. We adapted these to our setting in various ways, and incorporated them in *SolarView*. We test both with synthetic data, which has a known optimal embedding into two dimensions, and with data from OCLC’s WorldCat data set. The results show that our embedding competes well with these techniques and achieves low distortion in practice. We close in Section 7 with a discussion of our results and directions for possible future work.

2 RELATED WORK

Employing dimensionality reduction techniques [11], [13], [14] is a common approach to deal with many similarities at once. Typically, the results of dimensionality reduction are visualized as 2D scatterplots, in which the distances between points merely approximate the real distances. Controlled user studies have shown that users find it difficult to understand the placement of the points and how accurately the real distances have been preserved [9], and that there is a dependency between task and projection performance [13]. Users must also develop trust in the projection and representation to use the visualization confidently [10]. Stahnke *et al.* [11] propose a number of interaction techniques to aid users in interpreting the arrangements and errors of dimensionality reduction techniques. Their approach relies on a scatterplot type of display, which is enhanced by error halos and mouse over on a point to show the projection errors in relation to other points. The distributions of variable values for the original dimensions are shown in a sidebar, which does not scale well and also requires that these dimensions themselves are meaningful. Liu *et al.* [15] present a system to visually explore semantic relations in word embeddings in the context of natural language processing. They use dimension reduction to map the high-dimensional semantic space to two dimensions, and provide per-word distortion metrics and a view to display words and their direct neighbors in the high-dimensional

space to convey the inherent uncertainty introduced by dimension reduction.

Our approach of visualizing a focus entity and its closest neighbors is related to the visual exploration of query results. The Sparkler system [16] displays a rank score (similarity between a query and entities in a collection) in a radial visualization. The query term is in the center and the entities in the result are shown as dots arranged along a line where their distance to the center reflects similarity. Entities with an identical similarity score are positioned at the same distance to the center, and, to avoid overlap, they are spread along an arc. While this approach conveys the similarity of entities to a query term well, it is not suitable for our purpose, as we aim to show the similarities between all entities in the result.

A related approach for displaying distances between a focus and its close neighbors is the *buddy plot* from Alexander and Gleicher [17]. In a buddy plot, the focus is displayed to the left of a line segment, and its neighbors are displayed as discs on that line. These discs can encode some other variable using color, for example. As stated above, we want to show the similarity between all neighbors as well. For that a radial embedding is better suited, since the additional dimension allows for lower distortion. Moreover, our algorithm avoids overlap between entities.

PaperVis [18] also visualizes closest neighbors: a selected paper is shown at the center of a number of concentric discs. These discs contain papers related to the selected one. Papers are binned into ten categories, and every disc represents a bin. Visually, this approach closely resembles *SolarView*. However, the positioning algorithm for nodes in the output is completely different. As stated above, our technique preserves exact distances to the focus entity and attempts to minimize global distortion. PaperVis however, employs two positioning algorithms that both only partially attempt to minimize distortion. The first algorithm sorts papers within a bin and then places the papers spread out evenly across the ring, with a random starting angle. The second algorithm sorts all papers across all rings and then lays out the papers spread out evenly. This will result in potentially high distortion. PaperVis also allows papers to be on more than one ring within a bin, while *SolarView* does not; *SolarView* can bin entities to rings to create more visual structure in the output, while the binning by PaperVis stems from their characterization of distance between papers.

In theoretical research, dimensionality reduction techniques are often studied in the context of *metric embeddings*: mappings between metric spaces that aim to preserve distances. Of particular interest to our research is the work on metric embeddings into low-dimensional spaces. Computing embeddings with minimum distortion is NP-hard, even when embedding into a line or circle [19]. Even worse, in general the minimum distortion of an embedding into a line cannot be approximated within a factor polynomial in the number of elements [20]. Better approximation algorithms have been developed only for special input metrics, such as unweighted graphs or weighted tree metrics [20], [21]. Fomin *et al.* [22] give an exact algorithm for metric embeddings into the line that runs in $5^{n+o(n)}$ time. However, as mentioned in [22], this algorithm cannot easily be extended to embeddings onto a circle, and it is clearly not practical for

$n \geq 15$. Finally, Dhamdhere *et al.* [23] present a polynomial-time constant-factor approximation algorithm to minimize the average distortion of metric embeddings into the line. However, their definition of average distortion does not match ours (see Section 4) and is not as useful in practice.

We have a focus entity to which distances should not, or barely, be distorted, while all other distances should be distorted as little as possible. We propose to approach this by constructing a radial embedding of the entities. Radial layouts have been used successfully to visualize many kinds of data, such as hierarchical structures, relationships among entities, ranking of search results, and periodic data; see [24] for a comprehensive overview. Commonly, one circle is used in graph visualization and several concentric ones for trees. For the former case, much research effort has been put into arranging the nodes such that edge crossings and/or the sum of the edge lengths are minimized [25], [26]. While the node arrangement problem for minimizing the total edge length bears similarity to our proposed approach, our setting is different in that we have weighted edges (in essentially a fully connected weighted graph) and that we lay out the nodes on several concentric circles to reflect distances with respect to a focus entity.

3 PROBLEM ANALYSIS AND SCOPE

SolarView addresses the following general problem: We are given a number of entities in a high-dimensional metric space, one of which is marked as the focus entity. We wish to display these entities in the two-dimensional plane such that the distances in the plane resemble the original distances with as little distortion as possible.

Focus entity. Because it is generally impossible to avoid distortion, the distances in the output will most likely render a distorted view of the actual distances in the input. To offer users structure and provide a trustworthy core for the visualization, we eliminate distortion amongst pairs of entities involving the focus entity.

Examples of situations wherein a focus entity occurs are displaying search results (the focus being the query), detail-on-demand views where the local neighborhood of a selected entity is visualized, and navigation, where the current page has the focus and ancestors, recently visited and/or closely related pages may be in the neighborhood of that focus. We discuss concrete examples in Section 5.

Small neighborhood. We aim to make it easy to understand the local neighborhood of a selected entity, the focus. The idea is that we show the entities closest (most similar) to the focus, assuming that those are the most relevant to the user. Choosing this neighborhood to be not too large helps us achieve overall lower distortion and with that a better understanding of similarities. We hence assume that the number of entities in the input is small. We discuss the limits of this assumption in greater detail in Section 7.

Tasks. Using Brehmer and Munzner’s typology [27], we can define the abstract tasks we want to support.

- (1) SolarView produces, given a matrix representing high-dimensional distances between entities in a metric space, positions in 2D where entities must be plotted. Those

positions are derived from the input using the dimensionality reduction algorithm described in Section 4.

- (2) Users can discover ► browse ► compare entities, inspecting the distances between them. They can do so as SolarView encodes entities as points in the plane, allowing them to filter all entities to only the close neighbors of the focus, and select a neighbor to become the new focus, navigating to an updated view.
- (3) Users can browse entities with a similar distance to the focus and then compare distances between these entities independent of the focus entity.

These tasks apply to any data set consisting of entities for which distances or similarities are defined. Examples include semantic similarity of documents, travel time between geographic locations, and similarity of species of animals, given an appropriate similarity measure. Section 5 discusses concrete tasks on a philosophy dataset.

Note that because of our focus on the local neighborhood of a focus entity, we do not directly support *explore* type tasks. Thus, SolarView targets use cases where a focus entity is found by some other means (see the examples above). In particular, SolarView could be used as one view in a linked view context.

Efficiency and consistency. We want to support seamless navigation between different focus entities at interactive speeds. If the new focus entity is a neighbor of the previous focus entity, then the visualization should transition in an intuitive way, to maintain the mental map of the user [28].

Transparently communicating distortion. Whenever it is not possible to embed distances without distortion, the user must be able to easily inspect the distances to judge distortion accurately. The user must be able to reliably identify (dis-)similarities which are implied by position but in fact not correct. However, indicating all pairwise distances simultaneously for even a small set of entities will result in visual clutter and must hence be avoided.

4 COMPUTING RADIAL EMBEDDINGS

The overall goal of our visualization is to display similar entities close together and less similar entities further away from each other. More specifically, if $\mathcal{E}$ denotes the set of entities, then we want to compute a map $f: \mathcal{E} \rightarrow \mathbb{R}^2$, such that $|f(e_1) - f(e_2)|$ corresponds to the similarity between the entities $e_1, e_2 \in \mathcal{E}$ as much as possible. This is a dimensionality reduction problem, and standard dimensionality reduction techniques like *principal component analysis* or *multidimensional scaling* could be applied. However, in our setting, we have a special focus entity, and the distances to the focus entity should barely be distorted. Standard dimensionality reduction techniques give little to no control over which distances are distorted and they tend to distort many distances. Hence we cannot apply any such techniques directly and propose a novel type of low distortion radial embedding instead. We compare its performance to standard techniques (with postprocessing) in Section 6.

To analyze the quality of our technique, we use a quality measure from the area of *metric embeddings*. Metric embeddings are maps between two metric spaces, where the goal is to preserve distances as much as possible. More

formally, a *metric space* is a pair (X, d) , where X is the set of elements and $d: X \times X \rightarrow \mathbb{R}$ defines the distance between every two elements of X . A metric embedding between two metric spaces (X, d_1) and (Y, d_2) is a map $f: X \rightarrow Y$. The (multiplicative) *distortion* $\delta_f(x_1, x_2)$ of a metric embedding f for a pair of elements $x_1, x_2 \in X$ is defined as follows:

$$\delta_f(x_1, x_2) = \max\left(\frac{d_1(x_1, x_2)}{d_2(f(x_1), f(x_2))}, \frac{d_2(f(x_1), f(x_2))}{d_1(x_1, x_2)}\right) \quad (1)$$

Note that the distortion of a single pair of elements is at least 1. We require the embedding to be *non-contracting*, that is, $d_1(x_1, x_2) \leq d_2(f(x_1), f(x_2))$ for all $x_1, x_2 \in X$. With that restriction, the distortion can be simplified as follows:

$$\delta_f(x_1, x_2) = \frac{d_2(f(x_1), f(x_2))}{d_1(x_1, x_2)} \quad (2)$$

The most common quality measure for a metric embedding used in theoretical analysis is the maximum distortion: $\delta_{\max}(f) = \max_{x_1, x_2 \in X} \delta_f(x_1, x_2)$. However, the maximum distortion is a bottleneck metric. As such, from a practical perspective, the average distortion is a more suitable quality measure as it more closely captures visual perception:

$$\delta_{\text{avg}}(f) = \frac{2}{n(n-1)} \sum_{x_1, x_2 \in X \wedge x_1 \neq x_2} \delta_f(x_1, x_2) \quad (3)$$

Note that minimizing the average distortion is equivalent to minimizing the total distortion. We will use the average distortion of a metric embedding to analyze our technique.

Clearly, metric embeddings are strongly related to dimensionality reduction. We use a function $d: \mathcal{E} \times \mathcal{E} \rightarrow \mathbb{R}$ to denote the distances between entities (the opposite of similarities). The function d may not necessarily be a metric, but we can easily transform it into a metric, if so desired. Our approach will work in either case.

Bucketing. We sketched in Section 1.1 how our algorithm groups neighbor entities into k buckets—corresponding to k concentric circles centered on the focus—by their distance to the focus, and processes entities in the same bucket together.

From an algorithmic point of view there is a trade-off between having too many rings and too few rings. In the first case, entities which essentially have the same distance to the focus are not handled together, leading to possible conflicts. In the latter case, entities are handled together which do not in fact have a similar distance to the focus, so treating them as laying on the same circle is not accurate.

First and foremost, these buckets are an algorithmic tool to achieve an average low distortion at interactive speeds. However, the buckets and their circles can also be used to visually structure the output (see Section 5.1).

Algorithm. Our input is a set $\mathcal{E}$ of n entities and one special focus entity e^* , as well as a distance function d and a small constant number k , which denotes the number of concentric circles. Our embedding algorithm has several steps. Recall that neighbor entities that have a similar distance to the focus entity are restricted to lie essentially on the same circle around the focus entity. Their interactions have a strong influence on the distortion of the embedding. Our algorithm is hence handling them explicitly by first partitioning $\mathcal{E}$ into k groups $\mathcal{E}_1, \dots, \mathcal{E}_k$ according to their distance to e^* . For each of these groups we separately compute a metric embedding

Fig. 1: The output of our algorithm (TSP E) for $k = 5$.

onto a circle. To do so, we use a traveling salesman tour heuristic (see Section 4.1). These embeddings on circles are computed independently and the intra-circle distortion is independent of the rotation of the circles. In the second step we hence aim to minimize the inter-circle distortion without changing the intra-circle distortions by rotationally aligning the circles. Finally, we apply local optimizations to the complete entity placement to further reduce the global distortion (see Section 4.2). Figure 1 shows an example, all distances to the focus entity are correct.

Algorithm RADIALEMBED($\mathcal{E}, d$)

1. Partition $\mathcal{E}$ into groups $\mathcal{E}_1, \dots, \mathcal{E}_k$
2. **for** $i = 1$ **to** k
3. Embed $\mathcal{E}_i$ onto circle (see Section 4.1)
4. Rotationally align circle for $\mathcal{E}_i$ (see Section 4.2)
5. Optimize entity placement (see Section 4.2)

4.1 Metric embeddings onto a circle

An important part of our algorithm is a metric embedding with small average distortion of a set of entities onto a circle. Minimizing the distortion of metric embeddings is notoriously hard. In fact, minimizing the maximum distortion of metric embeddings onto a circle is NP-hard [19]. We can hence expect that minimizing the average distortion is also NP-hard, since minimization problems involving averages tend to be harder than optimization problems involving maxima. Hence it is highly unlikely that a polynomial time algorithm for our embedding problem exists.

Intuitively speaking, our goal is to place similar entities close together and to make a tour through all entities. It can be expected that the tour through all entities, as implied by a metric embedding on a circle, is short. Conversely, a short tour should put similar entities close together, for otherwise

Fig. 2: Embedding a set of points (b) in 2-dimensional Euclidean space onto a circle. The mapping is indicated using colors, the circle using black edges. (a) the TSP heuristic with high average distortion; entity pair (e_1, e_2) is close in Euclidean space, but far apart on the circle. (c) the optimal embedding; entity pairs that are close in Euclidean space remain close.

it cannot be short. Finding the shortest tour is a well-known problem called the *Traveling Salesman Problem* (TSP).

TSP is a famous NP-hard problem which can be solved optimally and efficiently for input sizes of more than a thousand points. This is the case since the classic *Integer Linear Program* (ILP) formulation for TSP is amenable to so-called branch-and-bound optimization. Unfortunately similar optimization techniques do not seem to perform efficiently enough to compute optimal metric embeddings on a circle, not even for small inputs of roughly 40 entities. We are hence turning towards a heuristic approach, which does use an efficient solution to TSP: we compute a traveling salesman tour of minimum length for the set of entities and use the resulting order of the entities to place them on the circle. We refer to this approach as the *TSP heuristic*.

The problem we tackle using the TSP heuristic bears significant similarities to that of *seriation*. Seriation is known under various names, including structuring of matrices, data reorganization, clustering of data arrays and rearrangement clustering [29]. Here rows (or columns) of a matrix are reordered in such a way that the similarity between neighboring rows is maximized. Equivalently, the distance between neighboring rows is minimized. The first and last row have only one neighbor however, while in our situation every entity has precisely two neighbors or is the only entity on a circle (the two neighbors can be the same entity). Various techniques are known for finding (a good approximation of) an optimal permutation of the rows. An overview can be found in [30]. Ankerst *et al.* [31] used a TSP tour to order dimensions in parallel coordinate plots and other visualization methods. Climer and Zhang [29] propose a technique closely related to our approach. They also use the TSP, but use dummy rows to, amongst others, avoid the problem of the first and last row having only one neighbor. In contrast, TSP is directly applicable in our situation.

The TSP is formally defined as follows: given a set of n cities and distances d_{ij} between cities, compute the shortest tour visiting all cities and returning to the first city. To compute the traveling salesman tour efficiently, we use the following formulation as an ILP, which is mathematical folklore—its first use can be traced back to 1954 [32]:

$$\text{minimize } \sum_{i,j} x_{ij} d_{ij} \quad (4)$$

$$\text{subject to } \sum_j x_{ij} = 2, \quad i \in \{1, \dots, n\} \quad (5)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad S \subset \{1, \dots, n\}, S \neq \emptyset \quad (6)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in \{1, \dots, n\} \quad (7)$$

This ILP formulation can be used to solve the TSP using a standard (Integer) Linear Programming solver, such as CPLEX.¹ Although seemingly complicated, implementing our technique with the help of such a solver is straightforward using the above formulation. Note that Equation (6) (which ensures that only one and not several smaller tours are returned) contains an exponential number of constraints. To achieve interactive running times we add these constraints lazily. That is, the ILP is solved without these constraints, and then we check which of them are violated by the solution (this can be computed efficiently). We repeatedly add these constraints and resolve until a proper solution is found. In practice this algorithm works very fast and can solve instances with approximately 40 entities at interactive speed (instances with up to 80 entities still solve within a second). See Section 6 for exact running times.

The TSP heuristic is intuitively very well suited to our embedding problem. Hence the question arises if it is in fact a constant factor approximation. Unfortunately this is not the case, see Figure 2. Here the optimal metric embedding has a significantly lower distortion than the embedding obtained by the TSP heuristic (in fact, the ratio between distortions cannot be bounded by a constant as n tends to infinity). Therefore we cannot make any theoretical claims about the efficacy of the TSP heuristic.

4.2 Optimizing entity placement

After we place all entities on the circles using the TSP heuristic, we need to combine the circles. For the individual circles rotation does not matter, but in combination it does. Since we optimize entity placement later, we do not need to be too exact. Therefore, we use the following simple approach. The circles are added from the inside out. The first circle can have any rotation. For the next circle we try 20 different rotations (and the mirrored versions) at regular angles and choose the rotation that minimizes the average

1. See <https://www.ibm.com/analytics/cplex-optimizer>.

Fig. 3: (a) the polar distance $D(e_1, e_2)$ between e_1 and e_2 . (b) the circular arc between the neighbors of e indicated in blue, and the forbidden regions indicated in red. (c) the set of circular arcs representing the positions where e may be placed.

distortion among the entities on the already placed circles. We repeat until all circles are placed.

The final step improves the placement of the entities to minimize the distortion between all of them. Since the TSP heuristic already computed a good order for the entities on each circle, we will maintain that order. To optimize the placement of the entities, we use the following iterative approach. For each entity we find the position (whilst maintaining the order) that minimizes the total distortion assuming that all other entities remain at the same position. We compute a new position for every entity, and repeat this process until the total distortion does not decrease significantly anymore. This approach is similar to the Gauss-Seidel method for solving a linear system of equations. Our experiments show that it converges quickly to a locally optimal solution.

We need to define how we measure the distance between two entities $e_1, e_2 \in \mathcal{E}$ in the radial embedding f . We could simply use the Euclidean distance $|f(e_1) - f(e_2)|$, but that does not clearly incorporate the important role of the different circles. Instead we use a distance measure that separates the radial distance and the angular distance. If e_1 has polar coordinates (r_1, ϕ_1) with respect to the focus entity e^* and e_2 has polar coordinates (r_2, ϕ_2) where $r_1 \leq r_2$, then we define the *polar distance* between $f(e_1)$ and $f(e_2)$ in the radial embedding as $D(e_1, e_2) = (r_2 - r_1) + r_1|\phi_1 - \phi_2|$. Intuitively, this is the length of the path that moves radially from the outer to the inner circle, and then along the inner circle to the other entity (see Figure 3 left).

To use this iterative approach, we first ensure that the embedding is non-contracting, for otherwise the distortion function is non-linear and hence more difficult to optimize. If needed, we do so by scaling the circles until the embedding is non-contracting. In fact, we scale up a little more (specifically, by a factor 1.5) to give the entities room to optimize their placement.

Now consider a single entity e that should be placed optimally. Since the order of the nodes must remain the same, the optimal placement must be within the circular arc defined by its neighbors (indicated in blue in Figure 3 middle). Furthermore, since the embedding must be non-contracting, e may not lie too close to another node. In particular, due to the polar distance D , this defines a circular arc (if not empty) for every other node $e' \neq e$ that acts as a

forbidden region for e (indicated in red in Figure 3 middle). Since the complement of a circular arc is also a circular arc, the set of valid positions for e is the intersection of a set of circular arcs. In general, this intersection can consist of several (at most $O(n)$) disjoint circular arcs (indicated in blue in Figure 3 right). Since the distortion function as given in Equation 2 is linear (and thus also the total or average distortion), the position with minimum distortion must lie on the boundary of the intersection of circular arcs. Since this set consists of only $O(n)$ positions, we can simply try all of them in $O(n^2)$ time and return the position with the lowest distortion. We can improve on this by sweeping through the positions in circular order and keeping track of the distortion function, which computes the optimal position of e in $O(n \log n)$ time.

We summarize the procedure in pseudocode below. The limit on the number of loop iterations can be tuned so that the optimization process can be pruned in interactive settings. In our experiments however, the optimization loop would usually only execute ten to twenty times.

Algorithm OPTIMIZEPLACEMENT($\mathcal{E}, d$)

1. **do**
2. $distortion = \sum_{e_i, e_j \in \mathcal{E}, e_i \neq e_j} \delta_f(e_i, e_j)$
3. **for** $i = 1$ **to** k **for each ring...**
4. **for each** $e \in \mathcal{E}_i$ **for each node on the ring...**
5. calculate intervals as illustrated in Figure 3
6. **for each** interval endpoint
7. place e at endpoint, record distortion
8. place e at position that minimizes distortion
9. $newDistortion = \sum_{e_i, e_j \in \mathcal{E}, e_i \neq e_j} \delta_f(e_i, e_j)$
10. **while** $\frac{distortion}{newDistortion} > 1$ **and** #loop iterations < 1000

5 A PROTOTYPE OF SOLARVIEW

We now describe a prototype implementation of SolarView that can be used to explore bibliographical records from OCLC's WorldCat [12] data set. A bibliographic record consists of several entities such as title, abstract, keywords, authors, journal, year of publication, and subjects. The context of such bibliographic entities in terms of their relation to other entities enables users to search through a meaningful semantic space. For example, a researcher might search for the term *kant*, and will obtain the philosopher Immanuel

Kant and closely related keywords (e.g., *transcendental* and *critique*), but also the author Hegel (who was influenced by Kant, but who had substantially different ideas) as search results. The task of the researcher is to identify that the semantic distance between the keywords and Kant is different from the semantic distance between Hegel and Kant. Another example is a search for a keyword in combination with a filtering for authors to obtain prominent scholars working on that topic. Similar tasks exist in interpreting the semantic space in natural language processing, in which users wish to explore relationships in word embeddings [15].

Our prototype uses ARIADNE [33], [34], a tool developed at OCLC Research, to search in WorldCat. ARIADNE computes distances (representing similarities) between bibliographic entities in a 600-dimensional semantic space. Within this space ARIADNE identifies a set of meaningful closest neighbors for a given focus entity.

SolarView is developed as a web application, and the prototype is available as an online demo.² The server side component of SolarView is responsible for computing the radial embedding, while the client side component of SolarView fetches data from ARIADNE and visualizes the resulting radial embedding. The client side component provides a simple search bar that can be used to search for entities, with additional controls for changing certain search parameters. The search term is passed on to ARIADNE, which returns a matrix of entities and their pairwise distances. This distance matrix is passed to the server side component where a radial embedding is computed. Finally, this embedding is then visualized on the client side as a vector graphic (Figure 4) using D3.js.

5.1 Design decisions

As explained in Section 3, we do not visualize the complete data set, but only display entities in the local neighborhood

2. See <http://glammap.net:8080/solarview>.

Fig. 4: SolarView: entities snapped to bucket circles. Users can click on any entity to focus it, or use the search bar.

of the focus entity. By design, ARIADNE returns only the most meaningful neighbors, which is typically a set of less than 50 entities. Given this number, we chose to bucket the search results on 5 rings. Users can keep only four to five objects in their visual short-term memory [35]. Thus, even when displaying only a few dozen entities, it is still challenging to compare many distances at a glance. The entities snap to the bucket circles by default to alleviate this issue. This “rounding” introduces only a small distortion (see Section 6) but we believe that these choices allow a good trade off between accurate distances and clarity of presentation. The validity of these assumptions will have to be tested through user studies in future work. The number of rings can be set by the user, see Figure 5, and the snapping to rings is optional. Furthermore, our algorithm supports any choice of rings; SolarView uses equidistant rings.

Encoding entities. ARIADNE assigns a label and a type to every bibliographic entity. The label is a human-readable identifier of the entity. ARIADNE knows the following types: 📖 words or phrases; 👤 subjects; 👤 authors; 🏢 publishers; 📄 citations; 📊 clusters, and 🌐 external items (e.g. journals/articles). Entities are represented as discs with an icon from the Font Awesome icon set as indicated, the online demo has a corresponding legend (see Figure 4). The colors group similar types of entities to provide visual consistency. Labels can be placed to the top, right, bottom or left of the corresponding disc. We place the labels one by one, avoiding overlap as much as possible.

Distance stars. We use so-called *distance stars* to give clear visual cues to the user to judge distortion. A distance star encodes the distances from one entity to its closest neighbors in the high-dimensional input space by both line thickness and color lightness (dark-thick means close and thin-light means far). Double encoding distances makes them easy to read [36]. Distance stars are shown when a user mouses over an entity (see Figure 6). We stack the lines in a distance

Fig. 5: SolarView supports a user defined number of bucket rings: output of our algorithm (TSP R, Section 4) for $k = 11$.

Fig. 6: Distance stars indicate distortion and are shown when a user mouses over an entity. Thick (and dark) edges indicate that entities should be close together.

star from thinnest to thickest to minimize occlusion and we apply edge casings to separate lines visually.

Distance stars indirectly represent the distortion between two entities as the discrepancy between the distance indicated by a distance star and the perceived distance between the two corresponding discs. Alternatively, we could directly encode the distortion instead of the distance (as to clearly indicate which pairs of entities are distorted), but we believe that distortion is a confusing concept to users. Distortion involves two distances, only one of which is actually visible, as well as the ratio between these two distances. Such ratios are difficult to interpret. We conducted a small informal study with two expert users working in philosophy which showed that (i) distortion indeed proved to be confusing, and (ii) the distance stars were considered to be a helpful tool. We let the users search for authors and topics that they were knowledgeable about, and then asked if they could interpret the resulting embedding and the distance stars. They used the distance stars to confirm that entities that were close together in the visualization were indeed semantically related. Furthermore, the distance stars proved to be helpful in obtaining a grouping of subfields or authors and similarities between them.

5.2 Navigation

A click on an entity will trigger a search for that entity, making it the new focus entity. The visualization should transition in an intuitive way, to maintain the mental map of the user [28]. We use the following animation to transition to the new view. First, entities that are not in the new embedding are faded out. Then, entities that are in both the old and new embeddings orbit around the focus entity towards their new positions. This is done by linearly interpolating

the polar coordinates of the entities. All entities move at the same rotational speed to make it easier to track them. When all entities have moved to the new positions, the new entities are faded in. To reduce visual clutter, we fade out the entity labels before we animate the positions. When the new entities are faded in, all labels reappear.

6 EXPERIMENTAL EVALUATION

In this section we experimentally evaluate our embedding algorithm and compare it to state-of-the-art dimensionality reduction and embedding techniques. None of these techniques are designed to control specific distances in the embedding; we hence adapted them to our setting in various ways. In Section 6.1 we describe the methods we tested against as well as our modifications in detail. In Section 6.2 we report the results of all methods applied to synthetic data which has a known optimal embedding in two dimensions. In Section 6.3 we compare all methods on real-world data from OCLC’s WorldCat data set. Distortion for all methods is measured using the polar distance defined in Section 4.

6.1 Methods

Below we describe the various methods we used in the evaluation. We denote each method or adaptation by an acronym that encodes the base dimensionality reduction technique and how it was adapted to produce the correct distance to the focus entity. We consider the following base dimensionality reduction techniques:

t-SNE. The t-SNE technique [37] uses probability distributions to map entities in high dimensions to points in 2D. The distances in the resulting space are deformed, in the sense that clusters of points are emphasized. Distances between points in the same cluster are relatively small, while distances between clusters are relatively large. We used an existing implementation in Java³.

MDS. Multidimensional scaling (MDS) is technically the definition of an optimization problem. However, the term MDS is often used in practice to refer to algorithms solving this problem, and thus is not uniquely defined. In this evaluation we use classical MDS [38], also known as Torgerson Scaling. This technique essentially uses principal component analysis (PCA) as a subroutine. MDS takes a matrix with dissimilarity values and produces a coordinate matrix that minimizes a *strain* function. In particular, we use MDSJ⁴.

FDL. Force-directed layout algorithms work by simulating a physical system with springs, that iteratively moves entities until tension is minimized [39]. We have implemented our own versions of these algorithms.

TSP. This is our new embedding algorithm (see Section 4). First of all, each of the above methods is used directly as is. However, except for TSP, the distances to the focus entity will not be correct. We therefore also evaluate a number of adaptations. In particular we employ several different variants of FDL. To avoid confusion, we denote the standard (unconstrained) FDL as UFDL. We also evaluate variants that, after each iteration of the FDL algorithm, (1)

3. See <https://github.com/lejon/T-SNE-Java>.

4. See <http://algo.uni-konstanz.de/software/mdsj>.

constrain entities to the rings (CRFDL), and (2) move entities radially to the correct distance to the focus entity (CEFDL). Furthermore, we also evaluate a hybrid method. Whereas FDL is initialized with points at random positions, we also use variants that use the output of TSP as initialization for FDL. In that case we prepend “TSP” to the acronym. We should note that, when used as initialization for FDL, we do not perform the rotational alignment or optimize the entity placement (that is, we skip line 4 and 5 in the Algorithm RADIAEMBED in Section 4).

Finally, we perform one of three types of postprocessing steps to correct the distances to the focus entity. The bold letter is appended to the acronym of the method.

Unconstrained. No postprocessing.

Exact. Entities are moved radially towards or away from the focus, to eliminate distortion with respect to it.

Ring. Entities are moved radially towards or away from the focus, to snap to the ring they are closest to.

Our algorithm from Section 4 is labeled TSP R, for example.

Some combinations of methods and postprocessing are not included in our experiments. For example, because TSP uses the rings, TSP U would give the exact same result as TSP R and is therefore left out. CRFDL E is inferior to CEFDL E and therefore left out as well.

6.2 Synthetic data

We tested all methods on a small synthetic two-dimensional data set where points are positioned in groups on concentric circles (see Figure 7 top left). This data set has a known optimal embedding into two dimensions, namely the input data set itself. It hence allows us to test how well the different methods preserve the structure in the input. Ideally, the points should be placed approximately on their input coordinates. Figure 7 gives a visual impression of the performance of all methods. The distances between the circles in the data set are greater than the distances between the circles in SolarView, causing TSP R to slightly rotate the circles relative to each other, since it is attempting to preserve the original input distances.

We can see that most methods represent the synthetic data set well, although slight variations exist between methods. The outlier is t-SNE, but that is to be expected since it does not focus on truthfully representing distances, but rather on emphasizing clusters.

6.3 Real-world data

We now compare all methods on real-world data from OCLC’s WorldCat [12] data set (see Section 5). Our dataset

Fig. 7: TSP R (right) properly represents point locations from a 2D synthetic data set (left), where groups of points are on concentric circles with different radii than those in SolarView. All methods are given for comparison in the table (bottom).

Fig. 8: Average distortion with 95% confidence intervals over 1 000 runs, for four different focus entities, with 5 rings (when applicable). Techniques are grouped by postprocessing measures (Unconstrained: no postprocessing; Exact: entities are moved radially towards or away from the focus, such that there is no distortion with respect to it; Ring: entities are moved radially towards or away from the focus to the ring they are closest to). Our algorithm described in Section 4 is called TSP.

contains about 150,000 authors, 200,000 key words, and 15,000 subjects extracted from 1.7 million articles in philosophy. We consider placement for a single focus entity at a time, and compare the distortion introduced by each of the methods. Inspired by the use case described in Section 5, we include the results of having *kant* as a focus entity. Additionally we consider *satisfies*, *tree*, and *theorem*.

The resulting distortions are shown in Figure 8. Because some of the techniques are non-deterministic, we run them repeatedly and consider the 95% confidence interval of the average distortion of all runs. We separately consider the overall average distortion, and the average distortion with respect to the focus entity only. Figure 9 gives a visual overview of the results of all methods on the terms *theorem*, and *kant*. Figure 12 in the appendix shows the same information for *satisfies* and *tree*. We also measured the runtimes of the different methods for computing the embedding with *satisfies* as focus entity. The results are shown in Figure 10.

Discussion. We first consider the results pertaining to distortion in Figure 8 and make a number of observations. First of all, the (best) average distortions differ for the different focus entities. For example, TSP R performs very well with *satisfies* as the focus, but with *tree*, CEFDL E and TSPCEFDL E have lower total distortion. This is caused by distance matrices being easier or harder to embed in 2D.

Second, as expected, the total distortion is usually lowest when the distances to the focus entity are not constrained. Constraining the distances to the focus entity increases the total distortion, but this effect is limited. Constraining the entities to rings adds some distortion in the distances to the focus entity, but this distortion is generally much smaller than in the unconstrained methods. Overall it seems that the limited increase in total distortion is a reasonable price to pay for distances to the focus entity that can be trusted.

Tradeoffs. If the goal is to put the entities on rings, as is the case in our application, then our embedding algorithm TSP appears to be the best choice, resulting in the least distortion in general. Indeed, TSP R is among the top in Figure 8, only outperformed with focus *tree*. However, if we want to enforce exact distances to the focus entity, then CEFDL appears to be the best choice overall. MDS seems to perform well in certain cases, but very poorly in other cases, making it not a reliable choice when embedding with a focus. Finally, t-SNE—and t-SNE R in particular—seems to perform poorly overall, but this is to be expected. Whereas the other dimensionality reduction techniques attempt to approximate the distances in the input distance matrix, t-SNE tries to detect and preserve clusters in the mapping to 2D. As t-SNE is not designed to minimize the distortion of the mapping, it is in some sense not fair to directly compare

Fig. 9: Example embeddings for the two focus entities *theorem* and *kant*. Observations: (i) t-SNE clearly clusters entities; (ii) FDL variants tend to use only part of the available space; (iii) different embeddings can obtain similar average distortion (see Figure 8), which is why we advocate tools—such as distance stars (Section 5.1)—that allow users to inspect distortion.

REFERENCES

- [1] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems*. Springer, 1991, pp. 199–233.
- [2] M. O. Ward, "XmdvTool: Integrating multiple methods for visualizing multivariate data," in *Proceedings of the Conference on Visualization*. IEEE Computer Society Press, 1994, pp. 326–333.
- [3] R. Beecham, C. Rooney, S. Meier, J. Dykes, A. Slingsby, C. Turkey, J. Wood, and B. Wong, "Faceted views of varying emphasis (FaVVEs): a framework for visualising multi-perspective small multiples," *Computer Graphics Forum*, vol. 35, no. 3, pp. 241–249, 2016.
- [4] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin press, 1983.
- [5] R. A. Becker, W. S. Cleveland, and M.-J. Shyu, "The visual design and control of trellis display," *Journal of Computational and Graphical Statistics*, vol. 5, no. 2, pp. 123–155, 1996.
- [6] E. R. Tufte and P. Graves-Morris, *The Visual Display of Quantitative Information*. Graphics press Cheshire, 1983, vol. 2.
- [7] S. van den Elzen and J. J. van Wijk, "Small multiples, large singles: A new approach for visual data exploration," *Computer Graphics Forum*, vol. 32, no. 3pt2, pp. 191–200, 2013.
- [8] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.
- [9] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences," in *Proceedings of the ACM BELIV workshop*. ACM, 2014, pp. 1–8.
- [10] J. Chuang, D. Ramage, C. D. Manning, and J. Heer, "Interpretation and trust: Designing model-driven visualizations for text analysis," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2012, pp. 443–452.
- [11] J. Stahnke, M. Dörk, B. Müller, and A. Thom, "Probing projections: interaction techniques for interpreting arrangements and errors of dimensionality reductions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 629–638, 2016.
- [12] OCLC, "WorldCat," <https://www.worldcat.org/>. Background information is found at <https://www.oclc.org/en/worldcat.html>.
- [13] R. Etemadpour, R. Motta, J. G. S. Paiva, R. Minghim, M. C. F. Oliveira, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 1, pp. 81–94, 2015.
- [14] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: a comparative review," *The Journal of Machine Learning Research*, vol. 10, pp. 1–41, 2009.
- [15] S. Liu, P. Bremer, J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci, "Visual exploration of semantic relationships in neural word embeddings," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 553–562, 2017.
- [16] S. Havre, E. Hetzler, K. Perrine, E. Jurrus, and N. Miller, "Interactive visualization of multiple query results," in *Proceedings of the IEEE Symposium on Information Visualization*, 2001, pp. 105–112.
- [17] E. Alexander and M. Gleicher, "Task-driven comparison of topic models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 320–329, 2016.
- [18] J.-K. Chou and C.-K. Yang, "Papervis: Literature review made easy," *Computer Graphics Forum*, vol. 30, no. 3, pp. 721–730, 2011.
- [19] P. Giannopoulos, R. Klein, C. Knauer, M. Kutz, and D. Marx, "Computing geometric minimum-dilation graphs is NP-hard," *International Journal of Computational Geometry & Applications*, vol. 20, no. 2, pp. 147–173, 2010.
- [20] M. Bădoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos, "Low-distortion embeddings of general metrics into the line," in *Proceedings of the Annual ACM Symposium on Theory of Computing*. ACM, 2005, pp. 225–233.
- [21] M. Bădoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Räcke, R. Ravi, and A. Sidiropoulos, "Approximation algorithms for low-distortion embeddings into low-dimensional spaces," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 119–128.
- [22] F. V. Fomin, D. Lokshtanov, and S. Saurabh, "An exact algorithm for minimum distortion embedding," *Theoretical Computer Science*, vol. 412, no. 29, pp. 3530–3536, 2011.
- [23] K. Dhamdhere, A. Gupta, and R. Ravi, "Approximation algorithms for minimizing average distortion," in *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science*, 2004, pp. 234–245.
- [24] G. M. Draper, Y. Livnat, and R. F. Riesenfeld, "A survey of radial methods for information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 759–776, 2009.
- [25] E. R. Gansner and Y. Koren, "Improved circular layouts," in *Proceedings of the International Symposium on Graph Drawing*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2006, vol. 4372, pp. 386–398.
- [26] Y. Koren and D. Harel, "A multi-scale algorithm for the linear arrangement problem," in *Revised Papers from the International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer-Verlag, 2002, pp. 296–309.
- [27] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.
- [28] J. Heer and G. Robertson, "Animated transitions in statistical data graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, 2007.
- [29] S. Climer and W. Zhang, "Take a walk and cluster genes: A TSP-based approach to optimal rearrangement clustering," in *Proceedings of the International Conference on Machine Learning*. ACM, 2004, p. 22.
- [30] H.-M. Wu, S. Tzeng, and C.-H. Chen, "Matrix visualization," in *Handbook of Data Visualization*. Springer, 2008, pp. 681–708.
- [31] M. Ankerst, S. Berchtold, and D. A. Keim, "Similarity clustering of dimensions for an enhanced visualization of multidimensional data," in *Proceedings of the IEEE Symposium on Information Visualization*, 1998, pp. 52–60.
- [32] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.
- [33] R. Koopman, S. Wang, and A. Scharnhorst, "Contextualization of topics - browsing through terms, authors, journals and cluster allocations," in *Proceedings of the International Society of Scientometrics and Informetrics Conference*, 2015.
- [34] R. Koopman, S. Wang, A. Scharnhorst, and G. Engleblenne, "Ariadne's thread: Interactive navigation in a world of networked information," in *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2015, pp. 1833–1838.
- [35] G. A. Alvarez and P. Cavanagh, "The capacity of visual short-term memory is set both by visual information load and by number of objects," *Psychological Science*, vol. 15, no. 2, pp. 106–111, 2004.
- [36] C. Ware, *Information Visualization: Perception for Design*, 3rd ed. Elsevier, 2012.
- [37] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [38] W. S. Torgerson, *Theory and methods of scaling*. Wiley, 1958.
- [39] P. Eades, "A heuristics for graph drawing," *Congressus numerantium*, vol. 42, pp. 146–160, 1984.

Thom Castermans received his Masters from the TU Eindhoven in 2015 and is now a PhD student at the same university. Part of a project titled *Visual Analytics for the World's Library Data*, he works in the area where algorithms, visualization and digital humanities meet. His research interests include geometric algorithms both in theory and in practice, and visual design.

Kevin Verbeek obtained his PhD at TU Eindhoven in 2012. From October 2012 to October 2014 Kevin was a postdoctoral researcher at UC Santa Barbara. Since November 2014 Kevin is an assistant professor in the Applied Geometric Algorithms group at TU Eindhoven. His main research interests lie within the area of computational geometry. Kevin is specialized in using theoretical techniques from computational geometry to solve real-world problems, mostly in the area of information visualization.

Shenghui Wang is a research scientist in OCLC EMEA office in Leiden, Netherlands. Her research activities include text and data mining, semantic embedding as well as linked data. Shenghui earned a Ph.D in Computer Science from the University of Manchester in 2007, a Master in Computer Application Technology at the University of Science and Technology of China in 2003, and a Bachelor in Computer Science at Anhui University in 2000.

Bettina Speckmann heads the Applied Geometric Algorithms (AGA) group at the Department of Mathematics and Computer Science of TU Eindhoven (the Netherlands). AGA mainly focuses on geometric algorithms for spatial data and applications of geometric algorithms in the areas of GIScience and Smart Mobility, geo-visualization, visual analytics, and e-Humanities. Bettina received her PhD from the University of British Columbia (Canada) in 2001. Afterwards she spent two years as a postdoc at the Institute for Theoretical Computer Science of ETH Zurich (Switzerland) before joining TU Eindhoven in 2003. Bettina was a member of both the Young Academy of the Royal Netherlands Academy of Arts and Sciences (2010-2015) and of the Global Young Academy (2011-2016). In 2011 she won the first Netherlands Prize for ICT Research.

Hein van den Berg Hein is assistant professor at the Department of Philosophy at the University of Amsterdam, Institute of Logic, Language and Computation. He specializes in the history of philosophy, the history of logic, and the history of biology. In addition, he is involved in various Digital Humanities projects.

Michel A. Westenberg is an associate professor of Visual Analytics for Healthcare at Eindhoven University of Technology (TU/e). He holds a PhD in mathematics and natural sciences (2001) from the University of Groningen, The Netherlands. Michel has been working at the TU/e Department of Mathematics and Computer Science since July 2008. He is the program director of the Data Science bachelor program, which is offered jointly with Tilburg University. Michel's research interest is data visualization and visual analytics in life sciences and healthcare, and he has past experience in image processing and computer modeling of the human visual system.

Arianna Betti is Professor and Chair of Philosophy of Language at the University of Amsterdam, Institute of Logic, Language and Computation. After studying historical and systematic aspects of ideas such as *axiom*, *truth* and *fact* (*Against facts*, MIT Press, 2015), she now endeavours to trace the development of ideas such as these with computational techniques in a strongly interdisciplinary setting. She did research at the universities of Krakow, Salzburg, Graz, Leiden, Warsaw, Melbourne, Lund and Gothenburg and held research grants among others from the European Research Council, the Italian CNR, the Dutch NWO, and CLARIN-NL. She has been a member of the Young Academy of the Dutch Royal Academy KNAW, of the Global Young Academy, and other international organisations dealing with research policy and topics such as science and society, open access and sustainability of research.

Rob Koopman is an applied research architect at the OCLC EMEA office Leiden, Netherlands. His research activities include semantic embedding, data mining and enhancing data quality. He started to work at OCLC (then Pica) in 1981 and designed and developed major parts of the EMEA systems. Since 2012 he does applied data science for OCLC.

APPENDIX

Fig. 12: Example embeddings for the two focus entities *satisfies* and *tree*.