Eurographics/ IEEE-VGTC Symposium on Visualization (2006)
Thomas Ertl, Ken Joy, and Beatriz Santos (Editors)

Affiliation Dynamics with an Application to Movie-Actor Biographies

Ulrik Brandes, Martin Hoefer †, and Christian Pich

Department of Computer & Information Science, University of Konstanz, Germany

Abstract

We propose a visualization approach for dynamic affiliation networks in which events are characterized by a set of descriptors. It uses a radial ripple metaphor to display the passing of time and conveys relations among the different constituents through appropriate layout. Our method is particularly suitable when assuming an egocentric perspective, and we illustrate it on movie-actor biographies.

Categories and Subject Descriptors (according to ACM CCS): information visualization, affiliation networks, time-dependent visualization

1. Introduction

Visual analysis of dynamic data is a challenging task, but offers unique possibilities for exploration. With the ever growing size and complexity of available data, tailored means of visualization for classes of such data are necessary to quickly keep track of and identify major changes and patterns in the temporal development.

We here address the visualization of a particular type of social networks, namely affiliation networks [WF94, Ch. 8]. Affiliation networks are represented as a bipartite graph in which the elements of one vertex set (the actors) are only connected to (affiliated with) the elements of the other vertex set (the events). It is assumed that the events are time-stamped and that there is a set of descriptors specifying their nature. Examples of such networks are interlocking directorates (directors affiliated with company boards distinguished by industry sectors), scientific publications (authors affiliated with publications characterized by keywords), or terrorist networks (terrorists affiliated with cells characterized by activity). We will use a movie database (actors affiliated with movies described by keywords) as our running example.

To animate the dynamics of an affiliation network, we use a concentric ripple metaphor, i.e. events are moving on a radial trajectory. Visual clustering is supported by determining the angle of a trajectory using barycentric coordinates in a space spanned by the descriptors. In addition, visual clues about the importance and influence of elements according to statistical or structural indicators are provided. Our visualizations thus provide an overview of the dynamics of the collection of actors, events, and descriptors that constitute the affiliation network. The design is particularly suited for visualizing egocentric affiliation networks, i.e. networks whose scope is defined by those events in which a particular actor (ego) appears.

This article is organized as follows. In Section 2, we provide some background and notation for the data model and its dynamics, and a discussion of related work. The visualization design and its layout are described in Section 3. An example application of our method to the Internet Movie Data Base is presented in Section 4, and in Section 5 we conclude with directions for future work.

2. Affiliation Networks

An *affiliation network* is represented by a hypergraph $N = (A, E)$ consisting of a set A of *actors*, and a set E of *events*, where $\emptyset \neq e \subseteq A$ for all $e \in E$. If $a \in e$ for some $a \in A$ and $e \in E$, actor a is called *affiliated* with or *incident* to event e.

† Author acknowledges support by DFG Research Training Group 1042 "Explorative Analysis and Visualization of Large Information Spaces

We will consider dynamic affiliation networks $N = (A, E; desc : E \to 2^D, T : E \to \mathbb{R}_0^+)$, in which events are labeled using a set D of *descriptors* and a time-stamp t_e. Such a network can, e.g., be represented as a tripartite graph with vertex set $V = A \uplus E \uplus D$ as shown in Fig. 1. The following

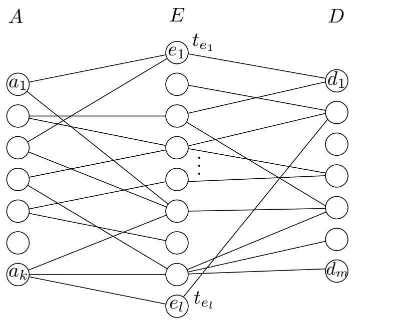

Figure 1: *The tripartite graph with partition sides A (set of actors), E (set of events) and D (set of descriptors) as well as time stamps t_e for every item $e \in E$.*

notion will be useful to define the similarity vertices, which in turn is used to determine their positions. Define the *neighborhood* N_w of a vertex $w \in V$ as the set of vertices incident to w. Note that, since there are no direct relations between A and D or within a single set, we have that $N_a \subseteq E$, $N_d \subseteq E$ and $N_e \subseteq A \cup D$ for all $a \in A$, $e \in E$ and $d \in D$.

2.1. Related Work

Social network visualization has recently been revitalized in the Nineties [BKR*99,Fre00], and quickly diversifies. Since affiliation networks, in particular, occur in many disguises, they are frequently not recognized as such and it is difficult to give a systematic account of their visualization. A straightforward, popular example is a web site called "They rule" (`www.theyrule.net`).

In [YFDH01] the authors propose a radial drawing method for large networks. The layout is centered around a specified node v, which is placed in the middle of the drawing. The graph is then treated as a tree rooted at v. It is drawn in a radial fashion on concentric circles, through which the distance to v becomes apparent. In addition, the authors describe smooth animation techniques for switching the root node. A radial drawing approach for social networks concentrating on structural properties is presented in [BKW03]. In addition there has been a lot of research on radial space-filling visualization techniques. As an example we mention [SZ00, YMR02], where methods for the display of hierarchical information are proposed. A different radial approach for visual correlation is considered in [LAMF05]. Visualization of dynamic social networks is discussed in [MMB05], and a prototype for visualizing online social networks is introduced by [HB05].

Closest to our approach in terms of visualization characteristics is the recent proposal of Appan et al. [AS05] for an egocentric layout of instant messenger data. The messages of a single participant are displayed in a circular fashion, where the participant (denoted *ego*) is placed in the center of the layout. Nodes representing her communication partners are located on a circle around it. Egos messages are displayed as nodes, and are moved from the center outwards to the corresponding partner. The partner nodes change their size depending on multitude and content of the communication with ego. In addition topcial information is displayed. Our method generalizes this approach in several directions. We propose a method for the display of general relational data that is not restricted to chat protocols. Thus, our approach lacks text-specific elements like the incorporation of topical information. In addition our layout method is more sophisticated because there is a more complex relational structure between the two vertex sets in the affiliation network. Events moved in the interior and their relations to the descriptors on the circle are of central interest. In the approach of Appan et al., communication partners on the outer circle are most important, and message nodes are used to simply indicate strength and frequency of communication.

3. Radial Layout

We use some rather natural intuitions about dynamic affiliation networks $N = (A, E; desc, T)$ to guide our layout design. We strive to place most recent events prominently in the center of the drawing. Older events are placed more peripherically, and it should be made clear which events are important to attract the center of attention. Older events like descriptors are used to provide a meaningful historical and conceptual context for the new developments. These goals are implemented in our layout as follows. At first, fixed positions for the descriptors are chosen on a *descriptor circle* that borders the layout to set up a conceptual context for the events. The time interval under consideration is divided into discrete fine-grained *ticks*, and for each tick a *time ripple* is introduced. Upon introduction, the time ripple is a point in the center of the drawing. As time progesses, the ripple grows larger in radius, but stays centered in the middle of the drawing. An event appearing in the layout remains fixed to the time ripple corresponding to its time stamp. Hence, as time progresses, events appear in the center of the layout and gradually wander to the outside. The relationships to descriptors and their position on the outer circle then determine where on the ripple an item is located. Formally, we model this using polar coodinates (r_e, β_e) for event $e \in E$. The radius r_e is given by the radius of the time ripple corresponding to t_e. The angle β_e is determined by the position of the descriptors from the neighboorhood N_e of e. In this way we place most recent events prominently in the center of the drawing, while older events at the border of the layout provide the historical context.

We apply the HITS algorithm for network analysis to determine importance and use this value to adjust label sizes for items and descriptors accordingly. Finally, each actor $a \in A$ is placed into the context of its neighborhood N_a. As coordinates we use the average polar coordinates of neighboring events.

3.1. Descriptor ordering

First and foremost the ordering of descriptors on the descriptor circle needs to be determined. In order to achieve a meaningful clustering of events, we allocate descriptors using a similarity measure based on the number of common neighboring events. This is natural, because in turn an event e is assigned angle β_e according to the neighboring descriptors as well. Our descriptor ordering should be designed that later on events can be placed close to historically and conceptually similar events and descriptors, while dissimilar events are well separated.

In our model, we say that two descriptor nodes d_1 and d_2 are similar if they have many common neighbors. Therefore, we use the *distance measure*

$$\delta(d_1, d_2) := 1 - 2\frac{|N_{d_1} \cap N_{d_2}|}{|N_{d_1}| + |N_{d_2}|}$$

to express dissimilarity between two nodes. This index is widely known as the Czekanowski-Dice-Sørensen similarity index [CS75] and used for classification and clustering in fields such as biology or sociology. It is designed such that $0 \leq \delta(d_1, d_2) \leq 1$, where $\delta(d_1, d_2) = 0$ if the two nodes have disjoint neighborhoods and $\delta(d_1, d_2) = 1$ if the neighborhood is equal.

For the descriptor circle a cyclic arrangement $\sigma : D \to \{1, \ldots, |D|\}$ of all nodes $d \in D$ must be found such that the sum of distances between consecutive nodes $d_{\sigma(i)}, d_{\sigma(i+1)}$ is minimized. This is equivalent to solving a traveling salesman problem (TSP) in a complete undirected graph $G_D = (D, D \times D)$, where distances are given by the Czekanowski-Dice-Sørensen index. TSP is known to be NP-complete, however, since our distance measure δ satisfies the triangle equality (i.e. $\delta(d_1, d_3) \leq \delta(d_1, d_2) + \delta(d_2, d_3)$ for all $d_1, d_2, d_3 \in D$), our descriptor TSP becomes metric, and heuristic approximation algorithms for metric TSP can be used. A simple yet powerful variant is constructing a minimum spanning tree (MST). A feasible tour is generated by walking around the MST, leaving out vertices that have already been visited. This yields a tour with length at most twice of the length of the optimal solution.

The best approximation method reaches a factor of 1.5 [Chr79] instead of 2; however, it is slightly more complicated to implement. Both heuristics are known to provide very good near-optimal solutions, so the the hardness of TSP does not really pose a problem – especially when dealing with instances of moderate size.

Once a tour $\left(d_{\sigma(1)}, \ldots, d_{\sigma(|D|)}, d_{\sigma(|D|+1)} = d_{\sigma(1)}\right)$ with length $|\sigma| := \sum_{i=1}^{|D|} \delta(d_{\sigma(i)}, d_{\sigma(i+1)})$ is found, it is projected to the perimeter of the descriptor circle, or equivalently, to the real interval $[-\pi, \pi)$, by

$$\gamma_i = \begin{cases} \gamma_{i-1} + \delta\left(d_{\sigma(i-1)}, d_{\sigma(i)}\right)\frac{2\pi}{|\sigma|}, & \text{if } 1 < i \leq |D|, \\ -\pi, & \text{otherwise.} \end{cases}$$

3.2. Radius

The *radius function* is a fundamental ingredient in the radial layout, as it determines the distance of objects to the center. Furthermore, it is useful for assigning influence values to all events. The events are to pop up in the center at the date of their time stamp. The movement to the border then decelerates with growing age, thus providing recent events with more space than older ones. However, events with finite age never reach the outermost descriptor circle. For these design principles the function

$$r(t) = \begin{cases} \frac{t}{t+k} & \text{if } t \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

is suitable to calculate the radius. The parameter $k > 0$ determines the time after which an event is in the middle between the center and the descriptor circle. More formally, the k determines t, for which $r(t) = \frac{1}{2}$. Fig. 2 shows a plot of the radius function for some values of k. Given an item $e \in E$,

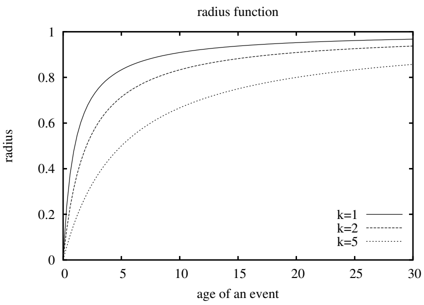

Figure 2: *The radius function with* $k \in \{1, 2, 5\}$.

the radius function determines the ripple with radius $r(t - t_e)$ on which e lies at time t. Figure 3 shows ripples for some multiples of k.

3.3. Event placement

The events are located such that the positions are related to the angles of neighboring descriptors. Given a set of k angles $\gamma_1, \ldots, \gamma_k$, the *average angle* γ is uniquely determined by

$$\sin\gamma = \frac{1}{\rho}\sum_{i=1}^{k}\sin\gamma_i, \qquad \cos\gamma = \frac{1}{\rho}\sum_{i=1}^{k}\cos\gamma_i,$$

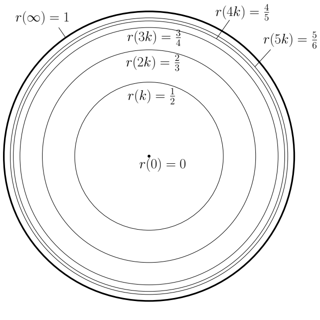

Figure 3: *Time ripple with various radii.*

where

$$\rho = \sqrt{\left(\sum_{i=1}^{k} \sin\gamma_i\right)^2 + \left(\sum_{i=1}^{k} \cos\gamma_i\right)^2}.$$

It can be obtained by computing

$$\theta = \begin{cases} \arccos\left(\frac{1}{\rho}\sum_{i=1}^{k}\cos\gamma_i\right), & \text{if } \sum_{i=1}^{k}\sin\gamma_i \geq 0, \\ -\arccos\left(\frac{1}{\rho}\sum_{i=1}^{k}\cos\gamma_i\right), & \text{otherwise.} \end{cases}$$

See Figure 4 for an example of an average angle. A straightforward way of placing an event is to use the average angle of all its descriptors. However, we incorporate information about time and distinctness into the contribution of descriptor angles. The average angle is biased with weights w_d for each descriptor d.

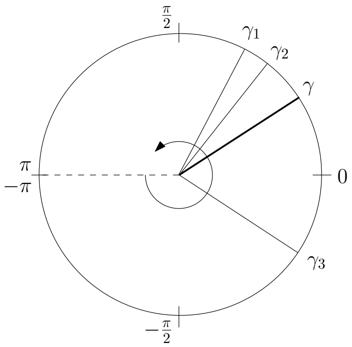

Figure 4: *The average angle γ of $\gamma_1, \gamma_2, \gamma_3$.*

To improve cluttering properties of the layout it is desirable to give discriminative descriptors more weight than others. This moves an event e into the direction of descriptors of N_e that are less frequently present in other neighborhoods, thus giving e a more distinct direction. This idea has been formulated in the area of information retrieval using the notion of *inverse document frequency*. It was introduced for keyword weighting [BR99] in document indexing. In our formal framework, the descriptor $d \in D$ is weighted with

$$w_d := \log \frac{|D|}{|N_d|},$$

giving rarely appearing descriptors more influence than frequent ones that are assumed to be less discriminative.

Using these weights, angles remain fixed once they are assigned. This may be sufficient in static settings, when our layout method is employed to display a snapshot of the evolution at a particular point in time. In the dynamic context, however, we want descriptors characterizing older events to be less influential than descriptors adjacent to recent events. To this end, we introduce the notion of temporal influence of an event e at time t dependent on the age of a event. As the radius is a convenient indicator of age, and the radius function r has the desirable property to map to $[0,1]$, we express the influence of event e as $1 - r(t - t_e)$. The *dynamic weight* of a descriptor d then becomes

$$w_d(t) := \log \frac{\sum_{e \in E} (1 - r(t - t_e))}{\sum_{e \in N_d} (1 - r(t - t_e))}.$$

3.4. Importance

When visualizing large data sets there is typically the need of providing more important or interesting items with more space than others. Therefore, it is necessary to be able to express *importance* for events and descriptors, and to create a meaningful space assignment by assigning font sizes. A widely-known approach is mutual reinforcement for importance computation. A node in a graph has a high importance score if its neighbors are considered important. We use this approach for the bipartite graph of events and descriptors. Events are considered important if they are associated with many important descriptors, and in turn, descriptors are given a high score if the events they describe are important.

Let $x \in [0,1]^E$ and $y \in [0,1]^D$ denote *importance vectors*. The importance x_e for each item $e \in E$ and y_d for descriptor $d \in D$ can be found by iteratively computing $x_e \leftarrow \sum_{d \in N_e} y_d$ for each $e \in E$ and $y_d \leftarrow \sum_{e \in N_d} x_e$ for each $d \in D$ and and normalizing x and y to unit length by dividing each entry by the Euclidean length $\left(\sum_{e \in E} x_e^2\right)^{1/2}$ or $\left(\sum_{d \in D} y_d^2\right)^{1/2}$, respectively. This is done iteratively until the maximum change of importance for any node in the iteration falls below a given threshold $\varepsilon > 0$. It is the essence of the well-known HITS algorithm [Kle99] for calculating *hubs and authorities*. In our layout these importance values are used to scale the font sizes of events and descriptors. In order to adjust the space occupied by the label, we use the square root of the values multiplied with a constant to derive the exact font sizes.

3.5. Actor placement

Actors are to be placed into the layout in a structured manner. It is important to support the visual presentation of clustering properties and temporal development of the affiliation network. Hence, an actor should always be located close to the important adjacent events that are present in the layout at the current point in time. We thus locate an actor a in a polar barycenter of the events in an adjusted neighborhood as follows. Let $N_a(t)$ be the *present neighborhood* at time t, i.e. the set of all adjacent events e with time stamp $t_e \geq t$. At time t actor a then gets located at polar coordinates $(r_a(t), \alpha_a(t))$, where the radius is derived by the average age of the present neighborhood:

$$r_a(t) = r\left(\frac{1}{|N_a(t)|}\sum_{e \in N_a(t)} t - t_e\right).$$

For the angle α_a a weighted average angle is used (see above). At time t we determine the weighted average angle of events in the present neighborhood $N_a(t)$. The weights are given by the normalized authority values associated with the events in $N_a(t)$.

3.6. Animation

In general we cannot assume a uniform distribution of time stamps. In addition, there are important applications, in which time stamps are rather coarse-grained (e.g. publication year). Our layout, however, should embed the course of events continuously. Hence, we need methods to distribute positioning changes onto the set of fine grained ticks in order to provide a smooth and stable animation.

For presentation, we will assume that time stamps are given as years. We consider the positions of an event e in two consecutive years t and $t+1$ as key frames. For the remaining ticks we interpolate between the two positions. Thus, at time $t+\tau$ $(0 \leq \tau < 1)$ the polar coordinates for an event e are

$$\big(r(t+\tau), \tau \cdot \beta_e(t+1) + (1-\tau) \cdot \beta_e(t).\big)$$

This results in a polar interpolation of the key frame positions. The event is moved circularly to its new position, which is natural in the given context, whereas a direct straight line through the inner of the circle would not comply with the overall goals of the layout. For actors the animation works accordingly.

4. Application: Actor Biographies

In this section we apply our method to a data set coming from the internet movie data base (IMDB, `imdb.com`). When exploring such huge datasets there is often interest in focusing on the curriculum vitae of one particular individual. Typically, one wants to know which movies have been most important for this actor. Likewise, the similarity of two given movies might be of interest, clustering properties, or whether it is possible to divide works and collaborators into meaningful groups. Hence, we use an *ego-centric* approach focusing on the biography of a single actor (denoted as *ego*).

For an ego we obtained all movies he has participated in with co-starring actors, and extracted keywords of the movies in the IMDB. A tripartite actor-movie-keyword graph was set up, where movies are events, and keywords are descriptors. The time stamp for a movie is the year, in which it appeared in the cinemas. For this graph we use the steps outlined above to create a keyword graph, from which we derive positions of keywords on the descriptor circle using the MST heuristic for TSP.

In the layout ego is located in the center and releases movies over time. Thus, in the course of the animation, the names of all the ego's movies pop up in the center and monotonically wander to the outside of a circular area, while their label size depends on their importance. Using weighted average angles, movies eventually cluster in similarity groups and show a topical and temporal clustering of an actor's biography. Actors are placed with weighted polar average coordinates in order to locate them close to the movies they appeared in. This forms a visual clustering of movies and actors in the conceptual space generated by keywords, and provides an overview of the development of an actors personal biography.

Figures 8 and 6 on the next pages show our layout for actor John Travolta in 1999. The graph consists of 5044 nodes – 3435 actors, 78 movies and 1531 keywords. For the layout we used $k = 10$ in the radius computation. Keywords d used only by one movie (with $|N_d| = 1$) were omitted because they are irrelevant for identifying topical connections of movies. Furthermore, there were simply too many keywords to include them into a readable, moderately-sized overview. After this adjustment 251 keywords remained in the graph. For the same reasons we only display actors that have appeared in at least three movies up to the current time t, i.e. for which $|N_a(t)| \geq 3$.

The IMDB contains entries for award shows and TV appearances that are identified as movies. Our method can provide a visual filter for these items clustering them on a horizontal line in the right part of Fig. 8. In addition, observe that the earlier dancing and music movies cluster in the upper right region. Later movies are located in the lower part of the image. Fig. 6 provides a closeup view on the music cluster, where correspondence of keywords, movies and actors can be verified. Furthermore importance values seem to roughly correspond to commonly perceived influence of Travoltas movies.

Fig. 7 shows screenshots of an example animation depicting the actor biography of Arnold Schwarzenegger. The graph consists of 4991 nodes – 3650 actors, 74 movies and 1267 keywords. Again, we removed keyword nodes of degree one leaving 173 keywords in the graph. The "Terminator" blockbusters and related items cluster in the lower part of the images. In addition, they are also assigned the largest font sizes, which correctly mirrors their perceived societal influence. Award shows appear on a horizontal line in the right half of the drawing, together with actors that co-appear with Arnold Schwarzenegger only in award shows and in no other movies.

Due to the increasing amount of events our layout is likely to experience visual cluttering in the border region. This is supported by the decreasing distance between time ripples. Here one can use dynamic importance values that decrease if the movies have passed a certain age. This results in smaller font sizes when the label reaches the border region. Another alternative would be to restrict the layout to the display of events from a certain time interval (e.g. for each time t to the interval $[t-10,t]$).

5. Conclusion

We have presented a novel layout method to display temporal development of tripartite relations derived from affiliation networks. A classic application of the method lies in the area of information retrieval and uses the vector-space model for document analysis [BDJ99]. In a collection of documents (events) each document contains a number of relevant terms (descriptors). It is natural to assume that documents have authors or publishers (actors), and that for each document temporal information is available about the time of writing. Using our visualization it is possible to observe the development of the collection in terms of topical structure or frequency of document creation. Furthermore, publishers and authors can be located in the topical environment of their documents. As a special case recent studies have focused on the analysis of bibliographic networks and co-authorship. Here our method can be used to visualize the existence and structure of research communities. In addition, it is possible to pick journals or authors as sources and keywords or journals as descriptors to get a topical or a publisher-related picture of the work under consideration.

There are a number of directions, in which our method can be extended. An interesting adjustment is to derive a dynamic importance measure for the events. The definition of present neighborhood can be extended to the whole network, i.e. we define the *current network* at time t as the tripartite subgraph induced by event nodes e with time stamp $t_e \geq t$. This network can then be used with the HITS algorithm calculating importance values for the present time t.

Another adjustment would be to include interactive elements like mouse-over effects. For instance, a mouse-over effect for an actor, event or keyword could be used to identify and highlight the neighborhood of this item. Also, methods to move the layout freely in time or to interactively zoom can be included. Finally, it is possible to extend our visualization into three dimensions. This, however, poses some new challenges when generating a good keyword distribution, as there is no easy reduction to TSP anymore.

References

[AS05] APPAN P., SUNDARAM H.: *Egocentric Analysis and visualization of instant messaging activity*. Tech. Rep. AME-TR-2005-19, Arizona State University, 2005.

[BDJ99] BERRY M., DRMAC Z., JESSUP E.: Matrices, vector spaces and information retrieval. *SIAM Review 40* (1999), 335–361.

[BKR*99] BRANDES U., KENIS P., RAAB J., SCHNEIDER V., WAGNER D.: Explorations into the visualization of policy networks. *Journal of Theoretical Politics 11*, 1 (1999), 75–106.

[BKW03] BRANDES U., KENIS P., WAGNER D.: Communicating centrality in policy network drawings. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 241–253.

[BR99] BAEZA-YATES R., RIBEIRO-NETO B.: *Modern Information Retrieval*. ACM Press, 1999.

[Chr79] CHRISTOFIDES N.: *Worst-case analysis of a new heuristic for the traveling salesman problem*. Tech. Rep. 388, Grad. School of Industrial Admin., CMU, 1979.

[CS75] CLIFFORD H., STEPHENSON W.: *Introduction into Numerical Classification*. Academic Press, 1975.

[Fre00] FREEMAN L.: Visualizing social networks. *Journal of Social Structure 1*, 1 (2000).

[HB05] HEER J., BOYD D.: Vizster: visualizing online social networks. In *Proc. IEEE Information Visualization* (2005), pp. 32–39.

[Kle99] KLEINBERG J.: Authoritative sources in a hyperlinked environment. *Journal of the ACM 46*, 5 (1999), 604–632.

[LAMF05] LIVNAT Y., AGUTTER J., MOON S., FORESTI S.: Visual correlation for situational awareness. In *Proc. IEEE Information Visualization* (2005), p. 13.

[MMB05] MOODY J., MCFARLAND D., BENDER-DEMOLL S.: Visualizing network dynamics. *American Journal of Sociology 110*, 4 (2005), 1206–1241.

[SZ00] STASKO J., ZHANG E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualization. In *Proc. IEEE Information Visualization* (2000), pp. 57–65.

[WF94] WASSERMAN S., FAUST K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[YFDH01] YEE K.-P., FISHER D., DHAMIJA R., HEARST M.: Animated exploration of dynamic graphs with radial layout. In *Proc. IEEE Information Visualization* (2001), pp. 43–50.

[YMR02] YANG J., M.WARD, RUNDENSTEINER E.: Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proc. IEEE Information Visualization* (2002), pp. 77–84.

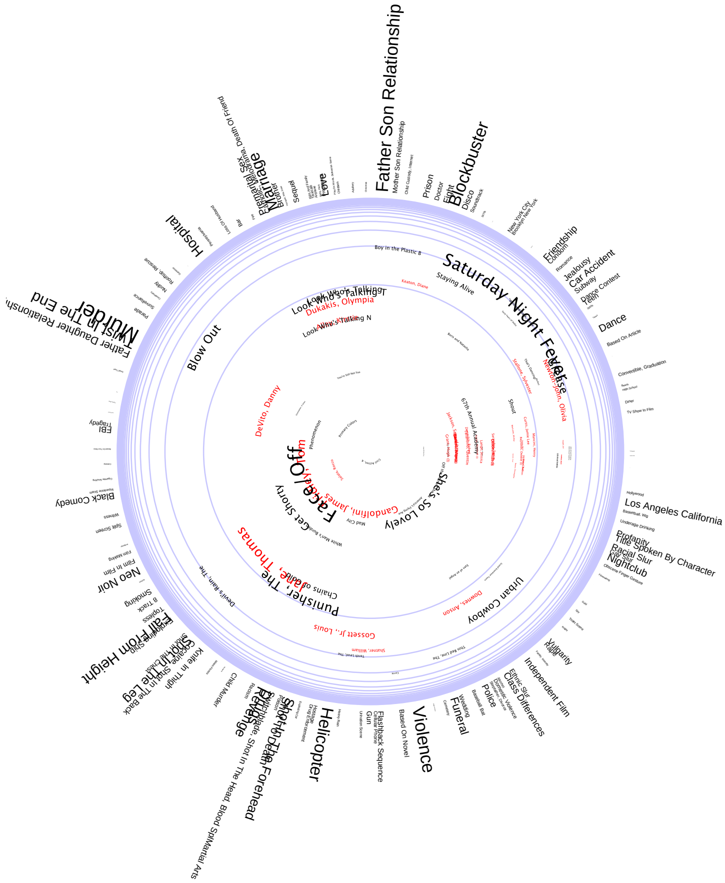

Figure 5: *A snspshot of John Travolta's movie biography layout in 1999. Dancing and music movies are concentrated in the upper right region, while his later movies tend downwards.*

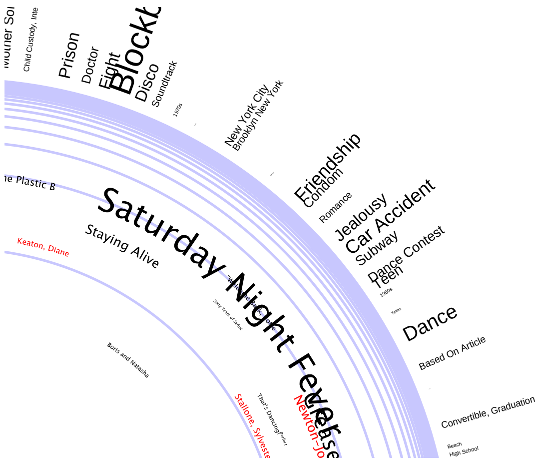

Figure 6: *A closeup view of John Travolta's dancing movie region, as indicated by some characteristic keywords. The font size illustrates that and Saturday Night Fever is associated with more important keywords than its sequel, Staying Alive. Due to co-appearance in those movies, Olivia Newton-John is placed to the same region, as well.*

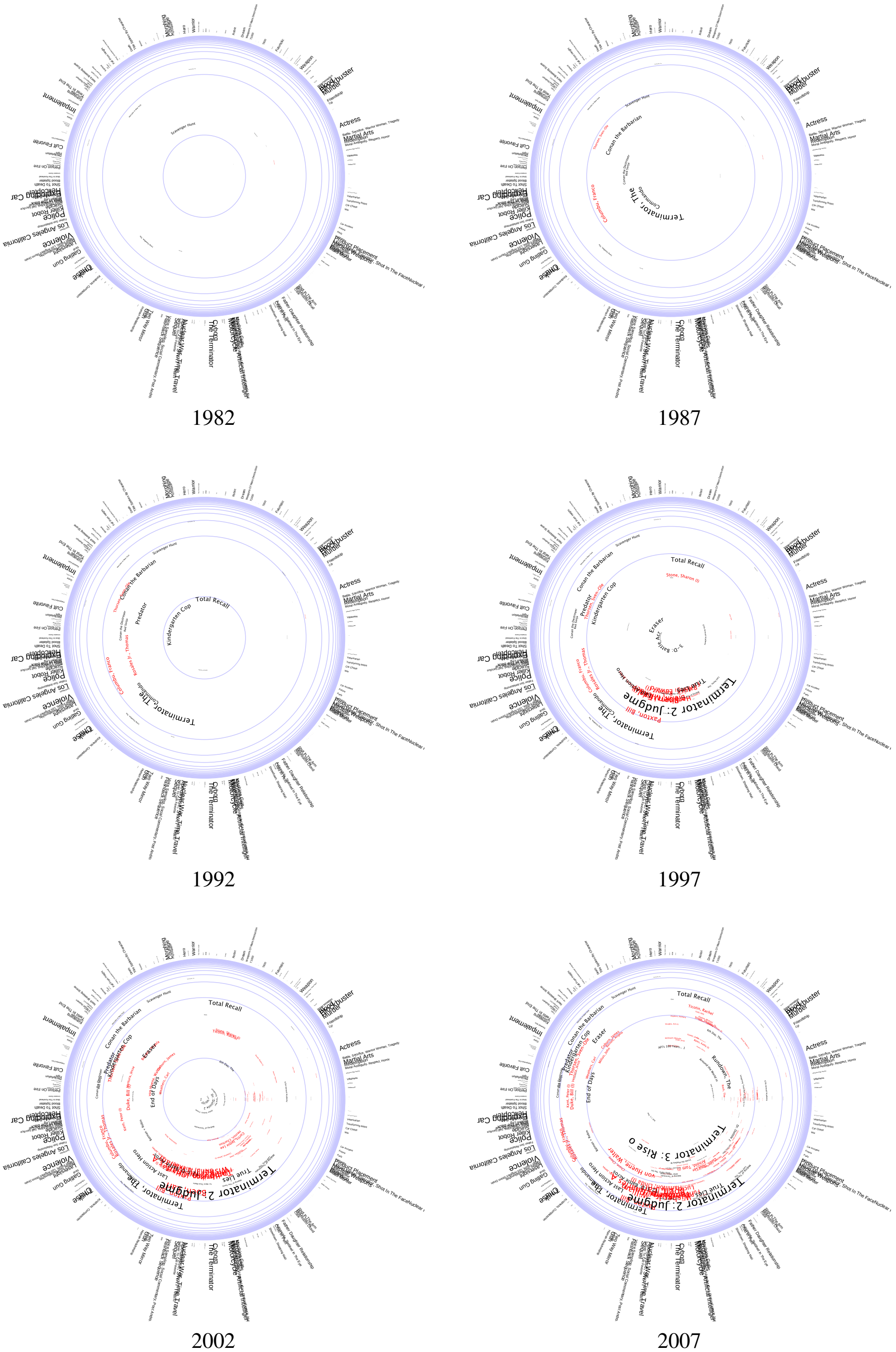

Figure 7: *Actor biography of Arnold Schwarzenegger. The snapshots are taken in intervals of five years, starting in 1982. In the lower region of the image "Terminator" movies are clustered.*