HAL
archives-ouvertes.fr

Interactive Visualization and Navigation of Web Search Results Revealing Community Structures and Bridges

Arnaud Sallaberry, Faraz Zaidi, C. Pich, Guy Melançon

▶ To cite this version:

HAL Id: inria-00538564
https://hal.inria.fr/inria-00538564

Submitted on 3 Dec 2010

Interactive Visualization and Navigation of Web Search Results Revealing Community Structures and Bridges

Arnaud Sallaberry
CNRS UMR 5800 LaBRI &
INRIA Bordeaux - Sud Ouest &
Pikko, Bordeaux, France
arnaud.sallaberry@labri.fr

Faraz Zaidi
CNRS UMR 5800 LaBRI &
INRIA Bordeaux - Sud Ouest
Bordeaux, France
faraz.zaidi@labri.fr

Christian Pich
Chair of Systems Design
ETH Zurich
Zurich, Switzerland
cpich@ethz.ch

Guy Melançon
CNRS UMR 5800 LaBRI &
INRIA Bordeaux - Sud Ouest
Bordeaux, France
guy.melancon@labri.fr

ABSTRACT

With the information overload on the Internet, organization and visualization of web search results so as to facilitate faster access to information is a necessity. The classical methods present search results as an ordered list of web pages ranked in terms of relevance to the searched topic. Users thus have to scan text snippets or navigate through various pages before finding the required information. In this paper we present an interactive visualization system for content analysis of web search results. The system combines a number of algorithms to present a novel layout methodology which helps users to analyze and navigate through a collection of web pages. We have tested this system with a number of data sets and have found it very useful for the exploration of data. Different case studies are presented based on searching different topics on Wikipedia through Exalead's search engine.

Keywords: Information Visualization, Web Search Results

Index Terms: E.1 [DATA STRUCTURES]: Graphs and networks—; H.2.8 [Database Applications]: Data mining—[H.5.3]: Web-based interaction—Group and Organization Interfaces

1 INTRODUCTION

The exponential increase in the information available on the web requires efficient organization and visualization systems to facilitate faster access to information [24]. A typical application of a visualization system in the context of information retrieval on the web is to visualize the search results of a query launched on a search engine [28]. Search engines such as Google, tend to return a long list of search results with titles, small images and short paragraphs. Users have to open each and every web page to assess its utility and relevance to the searched topic which can become tedious and unproductive [26]. As the information available on the web increases, there is an obvious need to organize and visualize these results [4].

Consider searching for the word *Jaguar* using Google Search Engine. Looking at the top seven results returned by Google (see Fig. 2), these results are distributed heterogeneously in the list, i.e. pages 1, 2, 3 and 6 are about the car manufacturing company called Jaguar, Pages 5 and 7 are about the animal also called *Jaguar* and Page 4 is about a super computer called *Jaguar*. If we look further in the list, we will find pages related to a software solution provider, a musical group, a guitar manufacturing company all having the name *Jaguar*. Ideally we would like to group the pages together based on their content so that the user can immediately realize that there are multiple themes related to the searched topic as shown in Fig. 1. To represent the contents of these groups, displaying noun phrases and keywords would allow users to glance contents
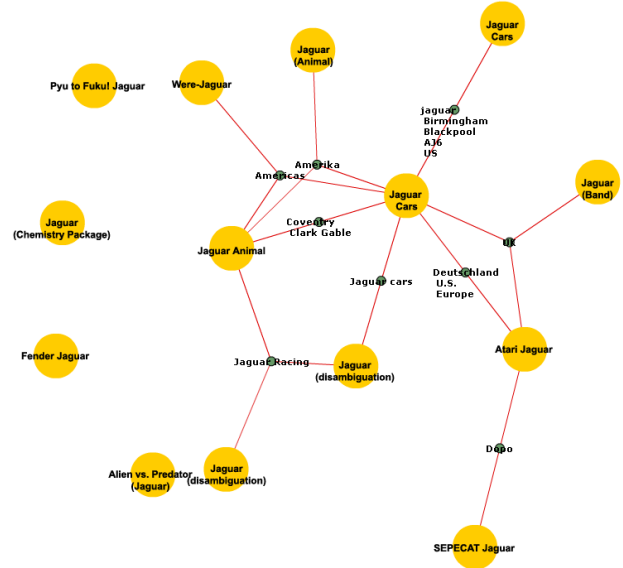

Figure 1: Visualizing Clusters and Bridges of the entire *jaguar* network. Distinct clusters (yellow nodes) clearly separate according to the different meanings of this keyword across web pages.

of search results without reading or scanning individual web pages and thus reduce their effort in locating relevant information [33].

To represent these keywords a co-occurrence network can be constructed where nodes represent keywords and edges link keywords if they appear together in a web page. These networks follow *small world* [32] and *scale free* [2] behavior, as shown by [12]. In the presence of these network properties, it is difficult to cluster these networks and present a challenging problem for the researchers. This problem was addressed in [34] where a clustering algorithm was proposed and applied to analyzing the contents of web pages collected through web browsing.

Visualization of clusters in small world and scale free networks remains a challenging problem [22]. Moreover representing web search results such that users can easily understand the content and navigate through the web pages remains an active area of research [27, 18]. In this paper we present a system that combines a previous clustering technique with a novel method to layout the clusters. This approach combines several established methods from the domain of information visualization and graph layouts to present web search results. These algorithms are adapted to cater the needs of underlying data and explicitly assist in its analysis, which in our case would be the different themes revolving around the searched topic. Moreover, the visualization also allows us to visualize the relationships between these themes by representing the
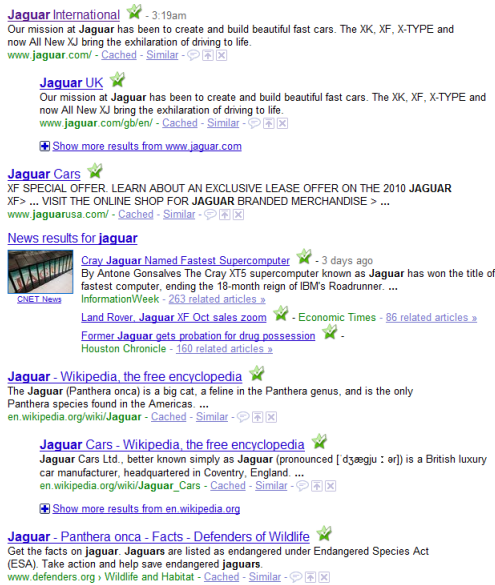
Figure 2: Screen Shot of the top seven Search Results returned by Google for the searched term *Jaguar*.

words that relate different clusters. We call these words *Bridges*.

The paper is organized as follows: The following section discusses the related work. In section 3 we briefly explain the clustering methodology and the notion of bridges as presented in an earlier work by the authors. Section 4 details the layout algorithm explaining the different interactors and navigators followed by different case studies presented in section 5. Finally we present conclusions and future research directions in section 6.

2 RELATED WORK

Different visualization systems for web search results can broadly be grouped into two categories: List Based Systems and Graphical Visualization systems. The list based systems keep the traditional ordered list visualization adding visual aids such as bolding words in the paragraphs [25] or clustering web pages and presenting a tree view [35, 33] along with the list. Graphical systems represent search results in a graphical environment where the visualization can either be 2D [28] or 3D [4]. The effectiveness of both list based systems and graphical systems has been investigated by different comparative studies but no formal proof exists and thus remains an open area of research [1]. In this paper we propose a Graphical Visualization System and give a brief account of the research done in visualizing search results as Graphical Visualization Systems.

WebSearchViz [28] is a graphical system that uses the metaphor of the solar system where the user query is placed at the center and the relevant pages placed around it as a function of the similarity to the user query. It uses a vector-based similarity measure to compute the degree of relevance but does not take into account the small world-scale free behavior of the keywords.

Kartoo [1] is a cartographic, visual meta-search engine. Kartoo labels the links between nodes in an attempt to give an idea about the kind of relationship between two connected nodes (sites), but these labels are frequently confusing and incorrect [6]. WebBrain [2] is another such utility on the web that helps users search and explore the web visually through a graphical representation. The search engine

---

[1] http://www.kartoo.com/
[2] http://www.webbrain.com

uses an egocentric [13] approach again placing the searched keyword at the center of the display area and the related web pages as a list on one side. The web pages are displayed at the bottom screen as we navigate through different searched keywords. The elements are not clustered which makes it difficult for the user to have an idea about the topics revolving around the key words searched.

LightHouse [26] is an information retrieval system that integrated both the list based and graphical based visualization to represent the clusters. The visualization uses spheres to represent web pages and two spheres overlap if they are semantically very close to each other. Although this is useful in case of a few web pages, but if many overlaps occur, it becomes difficult to visualize the web pages. And also, the user cannot see how the web pages are related to each other whereas this is possible through our system. The readers are recommended to read [28] which provides a good overview of the different visualization systems for web search results.

3 CLUSTERING METHODOLOGY

Most real world networks have both scale free and small world properties for example Internet Movie Database and the Coauthoring Network [21] or as in our case, the co-occurrence network. We know that due to the scale free property, it is difficult to cluster a network or visualize communities within the network structure [15]. This is because a few nodes have a very high degree and they create links between the underlying community structure hiding the communities as shown in Fig. 4(a). This figure is an example of a co-occurrence network which is drawn using a force directed algorithm [15] (see Sect. 5.2 for details). These algorithms are well suited to our problem as they put densely connected nodes closer to each other hence making it easier to locate the communities. But from the figure it is quite evident that in the presence of very high degree nodes it is very hard to identify different communities.

The clustering method earlier introduced in [34] was inspired from the fact that if a graph with both scale free and small world properties can somehow be reduced or transformed to a small world graph, clustering and visualization of such a graph would become easier. We proposed a two step process to cluster graphs:

3.1 Node Splitting

Typical behavior of scale free networks is that there are lots of nodes with low degree and a few nodes having very high degree. Fig. 3 shows the degree distribution of a scale free network taken from the example of Sect. 5.2.
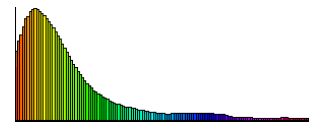

Figure 3: Typical Degree Distribution of a Scale Free Network showing the Long Tail behavior.

We proposed a node splitting method to change the scale free-small world networks to only small world networks. To reduce or eliminate the scale-freeness we proposed that if the high degree nodes (i.e. key words that are present in many web pages) are *split* in such a way that they are considered to be present only in exactly one web page, the degree of nodes (keywords) appearing in many web pages will decrease, thus leaving us with a network having only small world properties (see [34] for more details).

Fig. 4(a) shows a network obtained by extracting keywords from web pages obtained as a result of searching the topic CAC 40 on Wikipedia (see Sect. 5.2 for details). Fig. 4(b) shows the graph after node splitting. Quite clearly the network is more readable visually. Note that we keep track of the nodes split such that the
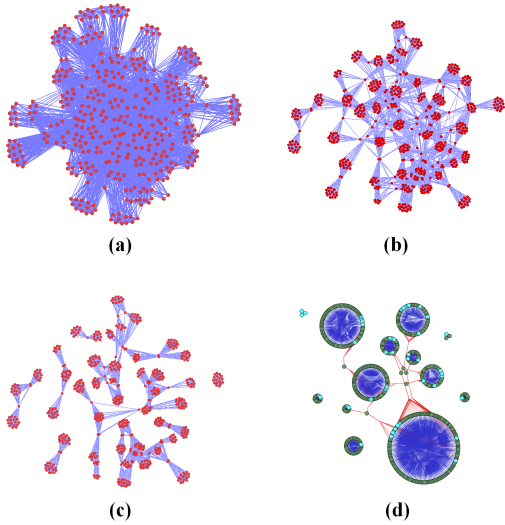
Figure 4: (a) Word-Word Graph constructed from browsing CAC 40 and related web pages (b) Graph after node splitting (c) Graph after removing bridges and identifying Clusters (d) Graph with Clusters and Bridges using the proposed visualization

final visualization enables the user to retrace the split nodes and thus avoiding any loss of information. Many systems that try to visualize complex systems use node or edge filtering to simplify the network [29, 30] whereas our approach keeps intact all the information and at the same time, produces a visualization which is easily readable. The readability is enhanced further by adjusting the visualization according to the clustering process, as is described in Sect. 4.

3.2 Iterative Removal of Nodes to Find Bridges and Clusters

The second step is to find the clusters and the nodes that connect these clusters to help understand how the clusters are related to each other. We used *Betweenness Centrality* [14] to identify the nodes that lie between communities of words representing the densely connected keywords.

For a graph $G(V,E)$ (where V is a set of nodes and E is a set of edges), this metric is defined as $BC(v) = \sum_{u \neq v \neq w \in V} \frac{\mu_{uw}(v)}{\mu_{uw}}$ where $\mu_{uw}(v)$ equals the number of shortest paths between two nodes u and $w \in V$ going through the node v and μ_{uw} equals the number of shortest paths between two nodes u and $w \in V$.

By definition, this metric assigns a high value to nodes that lie in the middle connecting many nodes. Thus it is well suited to find the words that are present in a number of documents and play the role of connecting keywords of different topics, i.e. they link web pages from different subjects. After identifying these words, we iteratively remove them temporarily, further simplifying the entire network to reveal disconnected communities as shown in Fig. 4(c). We then group the connected components as clusters.

The idea of using the betweenness centrality for clustering was proposed by [19], where certain edges were removed to find clusters. In contrast, since our goal is to understand how the clusters are connected to each other, we remove nodes instead of edges.

Once the clusters are found, the words that were initially split might find themselves in the same cluster. We only keep a single copy of a split node. A good example of a keyword that might get duplicated due to its high degree is the word *car* in case of the *Jaguar* example. Initially we might split it since it might have a high degree but once the different keywords concerning the *Jaguar*

Cars are grouped together in a single cluster, the multiple instances in a cluster of the word *car* can be removed keeping just a single node per cluster of this keyword. We reintroduce the nodes that were removed temporarily in the previous step and identify them as *bridges*. These bridges play an important role in understanding and analyzing the networks as shown in [34]. An important simplification step is to merge the bridges that connect the same clusters. As a result, these bridges actually become a set of keywords present in between two or more clusters. Fig. 4(d) shows the simplified form of the network obtained after applying the proposed clustering methodology and clearly shows a huge improvement in the visibility and readability of the network as compared to Fig. 4(a).

As a result of the proposed method, We obtain a bipartite graph $G(B,C,E)$ where B is a set of bridges, C is a set of clusters and E is a set of edges connecting nodes from set B and C. An edge exists between $b \in B$ and $c \in C$ if b appears on the same web page as at least one of the nodes (keywords) present in the cluster c.

4 LAYOUT ALGORITHM

Now that we have a set of nodes representing clusters and bridges as separate entities, a visualization system is required such that it helps the user analyze, understand and navigate through this graph. It is important to have a clear picture with clusters laid out such that bridges between clusters are well placed to give the user an idea of how the clusters are related to each other. Node and edge overlapping also needs to be taken into account as it is one of the fundamental criteria to produce readable drawings.

Foremost, we try to position the nodes of graph $G(B,C,E)$ in such a way that similar nodes are placed closely to each other, while dissimilar nodes are more separated geometrically. By using the lengths of shortest paths in the network, the proximity in a graph-theoretical sense serves as a proxy for topical similarity. In the graph drawing literature, this approach is often called "organic"; positions are computed by a simulation of physical forces or numerically minimizing an objective function [7].

In some preliminary experiments with different types of layout approaches and libraries, we found that existing implementations and traditional general-purpose layout algorithms are only of limited usefulness when particular aspects of the data are to be emphasized. Therefore, we adapt existing layout algorithms to produce a more dedicated layout method which explicitly takes into account the particular structure of our networks. It is more specific to the analytic perspective in our context and thus facilitates interpretation of *clusters* and *bridges*. Recall that the size of *clusters* is much bigger than that of *bridges* which needs to be accounted for when existing layout algorithms are used.

The layout process is discussed in the following sections and encompasses three major phases: first, the network is preprocessed in such a way that it can be fed to the optimization process, by hiding certain components of the network. Second, preliminary positions are computed. Third, a postprocessing step re-integrates elements hidden in the first step, before the final positions are determined.

Since typical instances are medium-sized (Exalead search engine extract about 650 concepts inducing about 4500 edges for each request of 50 web pages), computational efficiency is only a minor issue. In fact, the computational methods of the layout phases involved in our visualization scale to several thousands of objects, and no sophisticated data structures and fine-tuning efforts are required.

4.1 Preprocessing

4.1.1 Shortest-Path Distances

The computation of a layout for all nodes $v_1, \ldots, v_n$ is governed by the notion of similarity. Recall that we use the graph-theoretical distance to determine how far nodes should be apart from each other. Formally, given *target distances* d_{ij}, positions $p(1) = (x(1), y(1)), \ldots, p(n) = (x(n), y(n)) \in \mathbb{R}^2$ for every node in a

low-dimensional space, in our case two-dimensional, have to be found such that the resulting *Euclidean distances* $\|p(i) - p(j)\| = \sqrt{(x(j) - x(j))^2 + (y(i) - y(j))^2}$ correspond to these target distances as well as possible; how this is actually done is discussed further below in the second phase.

Generally, layouts constructed with this aim have frequently shown to be useful for uncovering symmetries and clusterings, which are not explicitly given, but are implicit in the data. The target distances, which can be interpreted as a dissimilarity measure in the node space, are computed by basic graph traversals.

Usually, we do not distinguish different types of weights of connections, and can thus use breadth-first searches from every node to construct. If edges are attributed general non-negative weights, the distance matrix may be built up by carrying out the algorithm of Dijkstra from each node, or by applying the matrix based algorithm of Floyd and Warshall; see, for example, [9]. In the next sections, let $D = (d_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}$ be the matrix of the target distances computed from $G(B, C, E)$.

4.1.2 Removing of Bridges for Layout of Clusters

Before laying out the clusters, we temporarily remove the bridges from $G(B, C, E)$. This is done so to ensure that the placement of bridges is done after the placement of clusters. Once we have placed the clusters, we can place the bridges between the clusters making it easy for the user to understand the relationship of clusters.

4.2 Layout Computation

4.2.1 Stress Minimization

Once the target dissimilarities have been computed, the coordinates should fulfill :

$$\|p(i) - p(j)\| \approx d_{ij} \tag{1}$$

for all pairs i, j as closely as possible where $i, j \in C$. This problem can be solved using *Multidimensional Scaling* (MDS) [5]. A robust approach is to quadratically penalize the deviation from the goal (1) for all possible pairs of nodes with the *stress* function

$$\sigma(p) = \sum_{i<j} w_{ij} \big(d_{ij} - \|p(i) - p(j)\|\big)^2 \tag{2}$$

which measures how far the current configuration $p = p(1), \ldots, p(n)$ is away from the given set of target distances $\{d_{ij}\}$. Here, we follow the de-facto standard weighting scheme used, most prominently, by [23] and set $w_{ij} = d_{ij}^{-2}$, which gives more influence to the representation of smaller distances and thus emphasizes more local structures.

Since there is no known closed-form solution for minimizing (2), it is optimized iteratively in a sequence of steps until the configuration is stable or the user is satisfied with the result. The following two subsections give more details on how to carry out the improvement step and what layout to start with.

4.2.2 Stress Majorization

The de-facto approach of computing the successive configuration with non-increasing stress is due to [10] and also popular for graph drawing [23, 17]. From a configuration $p^{[t]}$ at time t, an improved configuration $p^{[t+1]}$ is computed by local improvements.

For every object i coordinates $p(i)^{[t+1]}$ are updated using the current overall configuration $p^{[t]}$, while the other $n - 1$ objects are fixed. The update formula is

$$p^{[t+1]}(i) \leftarrow \frac{\sum_{j: j \neq i} w_{ij} \left(p^{[t]}(j) + s_{ij} \cdot (p^{[t]}(i) - p^{[t]}(j))\right)}{\sum_{j: j \neq i} w_{ij}} \tag{3}$$

where

$$s_{ij} = \begin{cases} \frac{d_{ij}}{\|p^{[t]}(i) - p^{[t]}(j)\|} & \text{if } \|p^{[t]}(i) - p^{[t]}(j)\| > 0 \\ 0 & \text{otherwise;} \end{cases} \tag{4}$$

this iterative updating is performed for each node and repeated until the configuration is stable and the stress cannot be reduced further significantly,

$$\frac{\sigma(p^{[t]}) - \sigma(p^{[t+1]})}{\sigma(p^{[t]})} < \varepsilon$$

for some $\varepsilon > 0$, e.g., $\varepsilon = 10^{-4}$. Alternatively, the computation may simply be terminated after a predefined number of steps or period of computation time. The generated sequence of layouts can be shown to have non-increasing stress, i.e.

$$\sigma(p^{[0]}) \geq \sigma(p^{[1]}) \geq \sigma(p^{[2]}) \geq \cdots \geq \sigma(p^{[t]}) \tag{5}$$

and to converge to a local minimum [11]. The motivation to use stress majorization is due to its low time complexity ($O(n^2)$) as the system requires a fast placement algorithm that can execute in real time, as compared to popular force directed algorithms such as [15] that have Time complexities in the order of ($O(n^3)$).

4.2.3 Initialization

A problem common to many iterative methods with a rugged objective function landscape is that the quality of the eventual solution is highly dependent on the initial solution $p^{[0]}$. Particularly, the majorization process discussed above notoriously gets stuck in bad local minima of the stress function, which correspond to undesired fold-overs and are very difficult to untangle [8].

To efficiently get an initial guess for a good layout, which is already nicely laid out on a global scale, we use *Classical MDS* [31], which is based on linear algebra. Let $D \in \mathbb{R}^{n \times n}$ be the distance matrix as above. Classical scaling is based on a matrix $B = (b_{ij})$ of *pseudo products*, having entries

$$b_{ij} = -\frac{1}{2} \left(d_{ij}^2 - \frac{1}{n} \sum_{s=1}^{n} d_{is}^2 - \frac{1}{n} \sum_{r=1}^{n} d_{rj}^2 + \frac{1}{n^2} \sum_{r,s=1}^{n} d_{rs}^2 \right). \tag{6}$$

Let $\lambda_1, \lambda_2 \in \mathbb{R}$ and $u_1, u_2 \in \mathbb{R}$ be the two largest eigenvalues and corresponding eigenvectors of B (note that B is symmetric and thus all eigenvalues and eigenvectors are real). Two-dimensional coordinates are then obtained by setting

$$x = \sqrt{\lambda_1} u_1, \quad y = \sqrt{\lambda_2} u_2 \tag{7}$$

as the (n-dimensional) coordinate vectors, which can be shown to be optimal [5] with respect to the mismatch between the "pseudo inner-products" b_{ij} derived from the distances and the inner products

$$\langle p(i), p(i) \rangle = x(i) * x(j) + y(i) * y(j) \tag{8}$$

of the current two-dimensional coordinates. For computing the required eigenvectors, a simple power iteration is sufficient [20].

4.3 Postprocessing

4.3.1 Re-Inserting Bridges

Once the clusters have been placed on the basis of their mutual distances, the bridges are re-inserted into the layout. We have found that it's intuitive and visually most effective to place bridges in the barycenter of the clusters they connect. Given a bridge b, which connects a set of neighboring clusters $N(b) = \{c \in C : (c, b) \in E\}$, the bridge is simply placed at
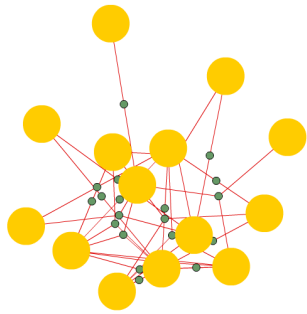
$$p(b) = \frac{1}{|N(b)|} \sum_{c \in N(b)} p(c) \tag{9}$$

Figure 5: Example of a graph after re-inserting bridges

Figure 6: Same example as in Fig. 5 after overlap removal

where $p(c)$ is the positions at which cluster c is centered. In the most frequent case of connecting two clusters, bridges are thus placed exactly in their middle; see Fig. 5.

An important remark about the placement of bridges is that they could have been considered as clusters and placed such that there are not many edge crossings between the edges connecting bridges. The problem with this approach is that the bridges cannot be placed between the clusters they belong to, thus creating a semantic inconsistency to read the final visualization. The method used currently allows edge crossings but preserves a mental map which is coherent with the semantics that bridge connecting two clusters should be put in between those clusters.

4.3.2 Overlap Removal

We display each cluster as a circle of area proportional to the number of nodes represented by it. If the layout approach described above is used unmodified and nodes are treated as if they were points having zero extension, it is likely to produce clutter because of overlapping nodes, even more so as clusters have variable non-zero sizes and are homogeneously distributed in the visualization area. Moreover, bridges should be visualized as bridges, i.e. properly located in between the clusters they connect, and should not be hidden by other clusters.

We adapt a simple, but effective approach using a modified stress model [16], which fits nicely into the stress majorization framework described above. In this approach, nodes (both clusters and bridges) are displayed as labeled rectangles, and overlaps are gradually eliminated by carefully moving pairs of overlapping nodes apart, while maintaining the topology of the overall configuration.

The main ingredient is an auxiliary *neighborhood graph* derived from the current layout. Particularly, we use the *Delaunay triangulation*, which can be efficiently computed, and which encompasses a planar graph $G^{\mathrm{DT}} = (V, E^{\mathrm{DT}})$ which is defined with respect to a two-dimensional configuration p.

Two nodes $i, j \in V$ with positions $p(i), p(j)$ are connected by an edge $\{i, j\} \in E^{\mathrm{DT}}$ in the Delaunay triangulation graph if and only if their *Voronoi* regions are adjacent, i.e. the sets of points in the plane which are closer to i and j, respectively, than to any other node in V. Note that G^{DT} is derived solely from the two-dimensional point set of node positions, and does not take into account the set of edges in the original graph $G = (V, E)$.

The actual overlap elimination is done iteratively:

- First, the Delaunay triangulation is computed from the current layout,

- Then, for every triangulation edge $\{i, j\} \in E^{\mathrm{DT}}$ the *overlap factor*

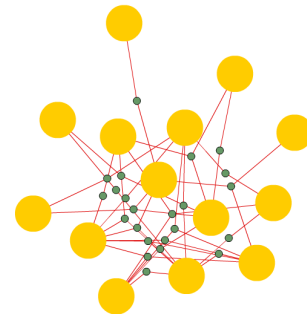$$t_{ij} = \max\left(\frac{a_i + a_j}{\|p(i) - p(j)\|}, 1\right) \quad (10)$$

is computed, where a_i, a_j determine the radius of the clusters i, j. The overlap factor determines by how much the current Euclidean distance has to be scaled up (1 if no overlap is present). The new desired distances are set to be

$$d_{ij}^{\mathrm{DT}} = s_{ij}^{\mathrm{DT}} \|p(i)^0 - p(j)^0\| \quad (11)$$

Here, s_{ij}^{DT} is a damping factor defined by $s_{ij}^{\mathrm{DT}} = \min\{s_{max}, t_{ij}\}$ where $s_{max} > 1$ is a constant damping factor which determines the maximum amount of overlap to be eliminated in one single iteration step.

- Finally, using the new desired distances, a modified proximity stress function

$$\sigma^{\mathrm{DT}}(p) = \sum_{\{i,j\} \in E^{\mathrm{DT}}} w_{ij} \left(d_{ij}^{\mathrm{DT}} - \|p(i) - p(j)\|\right)^2 \quad (12)$$

is minimized analogously to the stress majorization for computing the preliminary coordinates, as discussed above, using d_{ij}^{DT} and s_{ij}^{DT} instead of d_{ij} and s_{ij}.

These three steps are repeated until all overlaps are removed.

Another benefit of this algorithm is that the nodes spread out nicely and contribute to the overall readability of the final layout. This is due to the way the parameters are set and the results are shown in Fig. 6.

4.3.3 Navigation and Interaction

Visualization of clusters and bridges help users to build an overall picture of the search results. For a profound understanding and exploration of the returned results we propose different interactions to the user. Mouse rollover effect over a cluster displays the list of all the keywords present in the cluster as the tool tip (see Fig. 8). This helps the user to instantly identify what the cluster is about and take a decision about further exploring it. Moreover, the labels of the bridges are displayed which are useful to understand the relationships between clusters. Clicking on a cluster expands a cluster showing all the keywords in the cluster as nodes and clicking on it again, collapses the cluster (see Fig. 6). A Right click on a cluster displays the links to the web pages that are grouped in the cluster where the user can open any particular web page as shown in Fig. 7.

Apart from these interactions with the cluster, we can also interact with individual nodes within a cluster. Moving the mouse on a node, the label is displayed as the tool tip. We have used two colors to distinguish the split nodes (keywords) shown in Light Green Color as compared to the other shown in Dark Green. Upon clicking a split concept, all the instances of this keyword change their color to pink (see Fig.7) and increase the size so as to locate the
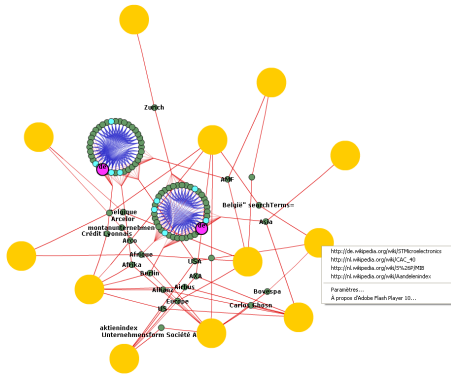
Figure 7: Same example as in Fig. 5 with expanded clusters

high degree nodes that were split in the node splitting step described earlier.

Expanding a cluster and displaying the nodes (keywords) in it, presents a new problem as the placement of the nodes is important to maintain a global perspective as well as to enable the user to explore a cluster in detail. The first thing we do is to replace the cluster with a circle which is bigger in size than the node representing the cluster. The radius of this new circle is proportional to the number of keywords in the cluster in this way, we are able to display all nodes of a cluster without missing any information. The next step is to place all the nodes of this cluster uniformly on the circumference of the circle. We first find a virtual position for each node, calculating the barycenter of the bridges with whom the node is connected. Then we calculate a circular order centered on the circle on which we are placing our nodes. This order is calculated using the virtual positions. Finally the nodes are placed following this order on the circumference of the circle. This method of placing nodes, based on *barycenter heuristic*, reduces edge crossings between nodes of a cluster and linked bridges (see Fig. 7). This layout constructed is called micro/macro and was proposed by [3]. Then, a new overlap removal step is performed using the new size of the cluster.

5 CASE STUDIES

The data sets that we have used in our experiments are collections of keywords found in web pages on Wikipedia. These web pages are returned as a search result when a query is launched on the Exalead search engine.

5.1 Search Word: *jaguar*

As a first example, we searched the word *jaguar*. As discussed in the section Introduction, this word represents completely different subjects like the Jaguar Cars, the animal etc. The top 50 web pages were used to extract 462 keywords connected through 4458 edges. The average clustering coefficient for this graph is 0.90 and the average path length is 2.42 representing its small world property.

Typically, semantic ambiguity of a word leads a search engine to return pages that are not semantically related, listing them in no particular order with respect to the possible meanings of the word *jaguar*. This is also present in the co-occurrence network, where keywords found in pages about Jaguar cars will be connected to keywords present in pages about the animal. The node splitting step identifies *jaguar* as a high degree node and disconnects keywords belonging to web pages related to cars or to the animal. This justifies our approach as nodes that have a high degree of connections need not to be grouped together in a single cluster as they are usually generic terms appearing with high frequency but not necessarily useful in terms of grouping related information together.

As a consequence, the visualization clearly positions different groups (cars, animals, video games, etc.) as distinct visual entities as shown in Fig. 1. Distinct clusters are placed apart and already indicate that the search results organize into groups of pages addressing different topics. Using the tool tip and browsing keywords contained in a cluster, users can quickly identify the underlying trends of the associated web pages. Fig. 8 illustrates this, showing keywords associated with pages dealing with a Japanese gag Manga named *Pyu to Fuku! Jaguar*.
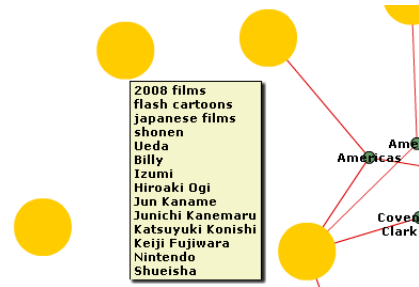

Figure 8: A tool tip allows to easily browse keywords of a cluster and figure out its intrinsic semantics.

Right-clicking on a cluster shows the URL of web pages associated with keywords. As Fig. 9 shows, in some cases the URLs already provide information about the underlying topic of a cluster. In our example, all pages obviously relate to different models of Jaguar cars.
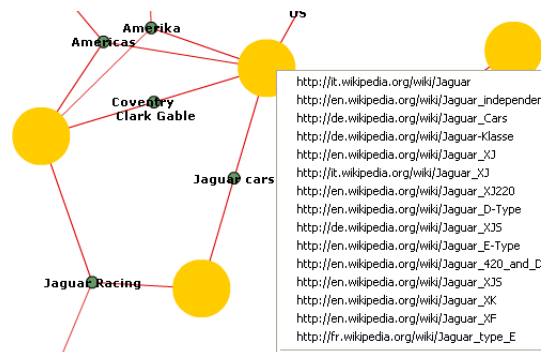

Figure 9: Right-clicking on a cluster reveals URL's of all web pages associated with keywords. In the example, URLs already indicate that the cluster gathers pages about Jaguar Cars.

5.2 Search Word: CAC40

Recall that the clustering algorithm identifies certain keywords as bridges. That is, although most keywords refer to different meanings and organize into distinct clusters, bridge keywords link clusters to one another. Hence, the overall graph separates into connected components providing a higher level organization scheme of the search results.

Our second example clearly shows how this acts at the data level. A search triggered from the keyword *CAC40* (an index associated with the top 40 companies listed in the Paris Stock Exchange – *Euronext Paris*) returned pages all related to finance. From a total of 50 pages, 556 keywords were extracted from these web pages and 4852 edges connecting these keywords. The average clustering

coefficient for this graph is 0.91 and the average path length is 2.56 representing the small world property of the graph.

The tool tip in Fig. 10 shows a group of web pages all related to the *financial crisis*. The associated cluster of keywords is connected to the CAC40 cluster (leftmost yellow node) through the bridges *Paris*, *AXA*, and *US Allianz*. *Paris* acts as a bridge connecting the *CAC40* and *Financial Crisis* clusters. This obviously suggests that along with the other global financial markets, *CAC40* was also affected. *AXA* is a French financial investment company in the CAC40 list. Fig. 11 shows the connection it induces between the cluster grouping (keywords of) web pages about *Euronext Paris*, the cluster *CAC40* and the cluster *Financial Crisis*. Since it is a financial investment company, it presumably is highly concerned with the financial crisis around the world, closely monitoring the Euronext Paris and being listed itself in the CAC40 index.
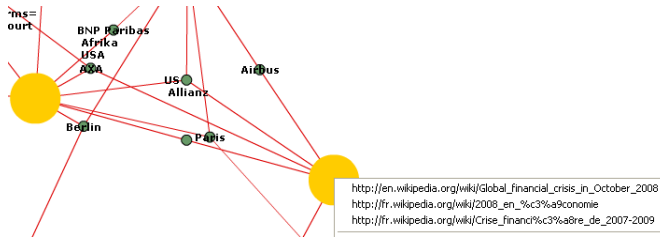

Figure 10: *AXA* acts as bridge inducing links between pages (keywords) related to the financial crisis and Euronext Paris of which AXA is a major actor. The list of web pages shown represents a cluster of web pages concerned with the financial crisis.
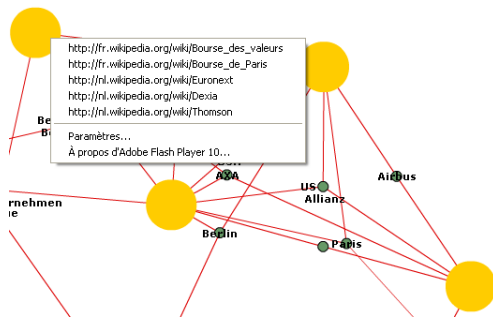

Figure 11: Changing focus to another cluster of Fig. 10 where pages related to the Paris Stock Exchange are shown.

5.3 Search Word: Hepburn

Finally, the third example represents a type of social network. We searched the word Hepburn which is a famous family name in Scotland. It is also quite frequent in some other areas of Europe, and we expect to find a social network of people belonging to that family. 554 keywords were extracted and 4636 edges connected them to form a co-occurrence network. The average clustering coefficient for this graph is 0.92 and the average path length is 2.51 representing the small world property of the graph.

Fig. 12 show the entire network obtained after applying the proposed visualization. This example again shows the effectiveness of the proposed method as the node splitting and bridge removal does not disconnect the clusters that are semantically related to each other. As shown in Fig. 13, we focus on four clusters that are connected to each other. These clusters are pages related to two

actresses Audrey Hepburn and Katherine Hepburn. They are not completely disconnected from the other clusters since they have the cinema as a common field. Let's take the example of the bridge *Tiffany's* which is extracted from a web page about a film called Breakfast at Tiffany's (1961) thus linking Audrey and Katherine with the cinema industry. Similarly, Audrey Hepburn appeared in a T.V commercial for the *KLM* airlines and Katherine played the role of *Ann Hamilton* in a film called Undercurrent. Thus the four clusters connected to each other are semantically related to each other. The clusters that appear apart from these clusters are pages related to politicians, writers belonging to the Hepburn family.
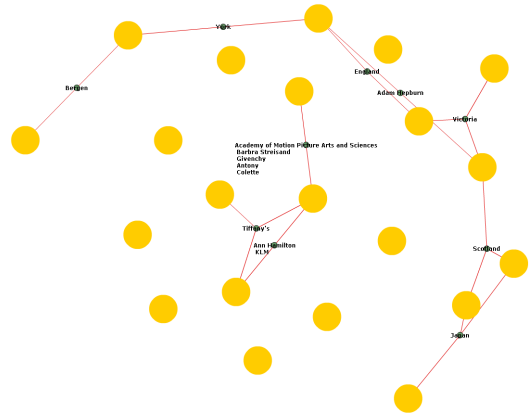

Figure 12: Visual Layout of Clusters and Bridges for the keyword *Hepburn* where a set of clusters are disconnected to other clusters.
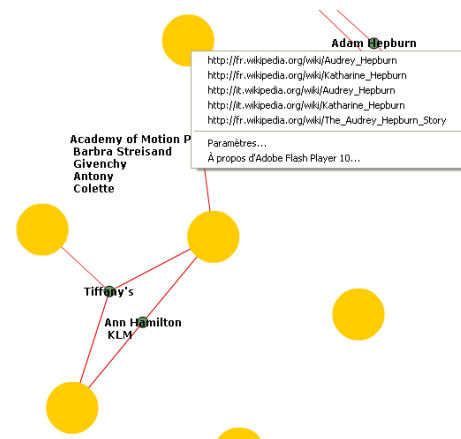

Figure 13: Focus on a connected set of clusters for the search keyword *Hepburn*.

6 CONCLUSION AND FUTURE WORK

We have presented a system to visualize and explore scale free and small world networks, as represented by clusters and bridges. Our approach is illustrated by several case studies which illustrate how our system is applied to the effective visualization of web search results. Our contribution is two-fold:

- Visualization: Our approach combines several established methods from information visualization and graph layout; this combination is dedicated to aesthetic criteria specific to the

data at hand and serves to explicitly reflect network analysis steps which are driven by the specific analytic perspective taken by the user.

- Implementation: Analysis and Clustering are integrated with the Visualization to produce a system which allows for interactive exploration and visual analysis of web search results.

The system was tested with small data sets as web search on a single topic does not require to evaluate hundreds of web pages at the same time. Similarly the size of documents was not very huge as web pages usually have a very limited size as compared to books. As future work, we would like to test the system to explore other large size networks having scale free and small world properties.

REFERENCES

[1] A. Aula. Enhancing the readability of search result summaries. In *Proc. of the HCI 2004: Design for Life, Leeds, UK.*, volume 2, 2004.

[2] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[3] M. Baur and U. Brandes. Multi-circular layout of micro/macro graphs. In S.-H. Hong, T. Nishizeki, and W. Quan, editors, *Graph Drawing*, volume 4875 of *Lecture Notes in Computer Science*, pages 255–267. Springer, 2007.

[4] N. Bonnel, V. Lemaire, A. Cotarmanac'h, and A. Morin. Effective organization and visualization of web search results. In *IMSA'06: Proceedings of the 24th IASTED international conference on Internet and multimedia systems and applications*, pages 209–216, Anaheim, CA, USA, 2006. ACTA Press.

[5] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer, 2nd edition, 2005.

[6] M. Boulos. The use of interactive graphical maps for browsing medical/health internet information resources. *International Journal of Health Geographics*, 2(1):1, 2003.

[7] U. Brandes. Drawing on physical analogies. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2001.

[8] U. Brandes and C. Pich. An experimental study on distance-based graph drawing. In I. G. Tollis and M. Patrignani, editors, *Proc. of the 16th International Symposium in Graph Drawing(GD'08)*, volume 5417 of *Lecture Notes in Computer Science*, pages 218–229, 2009.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[10] J. de Leeuw. Applications of convex analysis to multidimensional scaling. In J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, editors, *Recent Developments in Statistics*, pages 133–145. Amsterdam: North-Holland, 1977.

[11] J. de Leeuw. Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5(2):163–180, 1988.

[12] R. Ferrer I Cancho and R. V. Solé. The small world of human language. *Proc R Soc Lond B Biol Sci*, 268(1482):2261–2265, November 2001.

[13] D. Fisher. Using egocentric networks to understand communication. *IEEE Internet Computing*, 9(5):20–28, 2005.

[14] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

[15] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *GD '94: Proceedings of the DIMACS International Workshop on Graph Drawing*, pages 388–403, London, UK, 1995. Springer-Verlag.

[16] E. R. Gansner and Y. Hu. Efficient node overlap removal using a proximity stress model. In I. G. Tollis and M. Patrignani, editors, *Proceedings of the 16th International Symposium in Graph Drawing (GD'08)*, volume 5417 of *Lecture Notes in Computer Science*, pages 206–217, 2009.

[17] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In J. Pach, editor, *Proceedings of the 12th International Symposium on Graph Drawing (GD '04)*, volume 3383 of *Lecture Notes in Computer Science*, pages 285–295, 2004.

[18] J. Gao and W. Lai. Visualizing blogsphere using content based clusters. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, volume 1, pages 832–835, Dec. 2008.

[19] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:8271–8276, 2002.

[20] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[21] J.-L. Guillaume and M. Latapy. Bipartite graphs as models of complex networks. In *Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN), LNCS*, volume 1, pages 127–139, 2004.

[22] Y. Jia, J. Hoberock, M. Garland, and J. Hart. On the visualization of social and other scale-free networks. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1285–1292, 2008.

[23] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.

[24] D. A. Keim. Information visualization and visual data mining. *IEEE Transcations on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[25] M. Kickmeier and D. Albert. The effects of scanability on information search: An online experiment. *Proc. Volume 2 of the HCI 2003: Designing for Society*, 2:33–36, 2003.

[26] A. Leuski and J. Allan. Lighthouse: showing the way to relevant information. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on Information Visualization*, pages 125–129. IEEE Computer Society, 2000.

[27] T. M. Mann and H. Reiterer. Evaluation of different visualizations of web search results. In *DEXA '00: Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, page 586, Washington, DC, USA, 2000. IEEE Computer Society.

[28] T. Nguyen and J. Zhang. A novel visualization model for web search results. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):981–988, Sept.-Oct. 2006.

[29] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, Washington, DC, USA, 1996. IEEE Computer Society.

[30] I. Subasic and B. Berendt. Web mining for understanding stories through graph visualisation. pages 570–579, Dec. 2008.

[31] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.

[32] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[33] Y.-f. B. Wu, L. Shankar, and X. Chen. Finding more useful information faster from web search results. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 568–571, New York, NY, USA, 2003. ACM.

[34] F. Zaidi, A. Sallaberry, and G. Melançon. Revealing hidden community structures and identifying bridges in complex networks: An application to analyzing contents of web pages for browsing. In *IEEE/WIC/ACM WI-IAT '09*, pages 198–205, 2009.

[35] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 210–217, New York,USA, 2004. ACM.

---

[3] http://www.pikko-software.com/
[4] http://www.exalead.com/search/wikipedia/