Noname manuscript No.
(will be inserted by the editor)

A Multimedia Document Browser Based on Multilayer Networks

Benjamin Renoust[*,3,4] · **Haolin Ren**[*,1,2] · **Guy Melançon**[2] · **Marie-Luce Viaud**[1] · **Shin'ichi Satoh**[4]

Received: date / Accepted: date

Abstract Querying and retrieving relevant information still remains a difficult task, one with a relatively high cognitive cost for users, who usually focus only on the first few pages of results. This issue drives effort to support the exploration of search results through clustering and visualization. This paper contributes to this challenge by providing a visual analytics system that is designed to support search tasks in multimedia document archives. The system provides complex querying, semantic overviews of time, and visual, and textual concepts combined with analysis. All search tasks are supported with linked-highlighting and leapfrog interactions. This is made possible all in a single data structure thanks to multilayer network modelling.

Keywords Multimedia analytics · Visual analytics · Search · Browser · Multilayer networks

1 Introduction

Querying and retrieving relevant information still remains a difficult task, one with a relatively high cognitive cost for users [5]. While relevance ranking (such as page ranking [56] [12] [6] based on network modeling) is a powerful approach to extract a set of interesting pages, studies have shown that users usually focus on the first few pages of results [69] [70] (a behavior that barely changed over time [68]). Semantic ambiguity remains, challenging the information retrieval workflow, from extraction down to restitution to users [51] [36], for which context is key to disambiguation. This has been the focus of the Search Result Clustering (SRC) process in its last two steps: *labeling* and *visualization* [14].

[*] *Corresponding authors: renoust@ids.osaka-u.ac.jp, hren@ina.fr*
[1] French National Audiovisual Institute (INA), Paris, France
[2] University of Bordeaux, LaBRI CNRS UMR 5800, Bordeaux, France
[3] Osaka University, Institute for Datability Science (IDS), Osaka, Japan
[4] National Institute of Informatics (NII), Tokyo, Japan

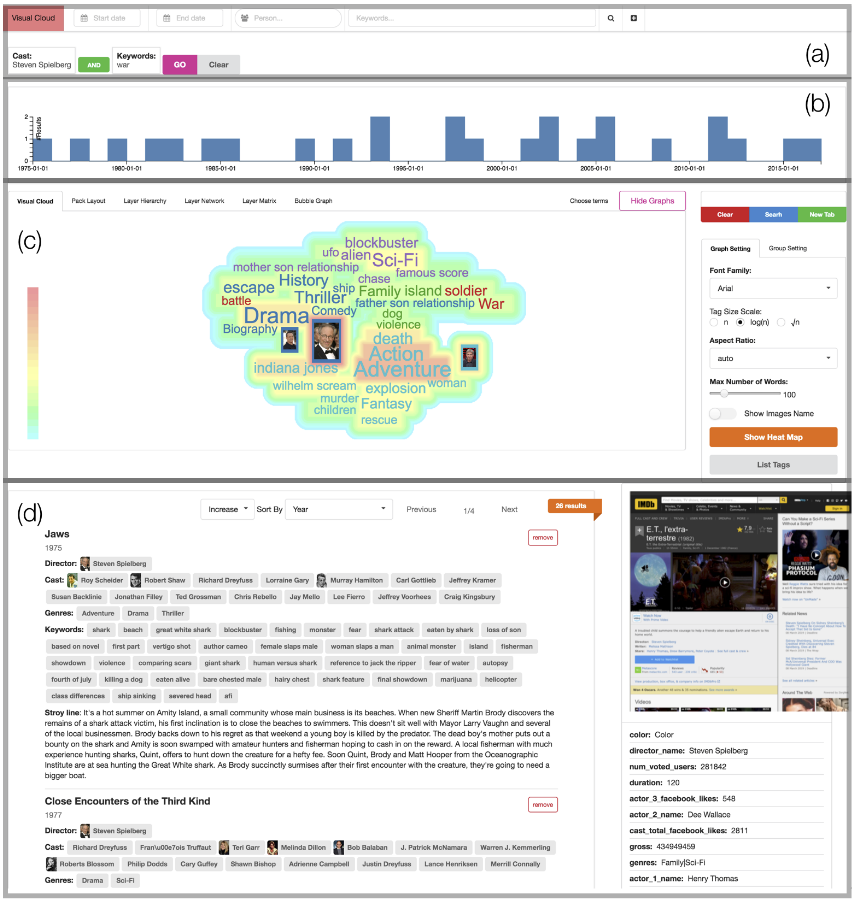


Fig. 1 Overview of the main elements of our search interface (a) Search bar with Boolean search query (b) Time overview with a time-line (c) Semantic overview with our visual cloud (d) Search results with details.

This has stimulated the study of strategies adopted by users to find their way into the information space [44] and information visualization soon responded to tackle this task [2] [76]. In particular, tag clouds can give an overview of this semantic space and support exploratory tasks [66]. The visual inspection of results and even snippets of the first few most relevant results cannot allow users to build a proper mental map of this space.

Visual analytics of multimedia data (or *multimedia analytics* [15]) proposes to extract high-level representations and support high-end analysis of multimedia concepts. Not limited to the textual information, the analysis of video content itself matters, due to the impact of images to the viewers [7]. Advance in computer vision now allows the extraction of visual semantic concepts, which can be used to index video documents in an archive [27] [38].

The search process may be divided into different tasks: *look-up*, *browse*, *locate* and *explore* [11]. It is difficult for a search engine to address them all at once, but we want to address this problem through proper visual representation and interaction.

We propose to give access to the contextual coverage of the results of a user query in form of a multimedia analytics system designed to address each of the search tasks. Our system enables complex searches and delivers results, while being supported by a visual map of the context formed by the result space. This context is built from temporal information, visual information, and textual information.

1.1 Multilayer Networks

Modeling search results under the form of a graph has been a successful approach to search engines [56] [12]. The multilayer model [32] extends the matrix representation of a network to a tensor formalization, and aims at modeling multiple types of relationships [13]. In the case of modeling multilayer document groups [61], the nodes of the multilayer network represent the documents, and each layer of links is a semantic concept shared between documents.

Multilayer networks have been used to support retrieval [40] and study the cohesion within groups of documents [61]. Cohesion is measured by observing the overlap of layers across documents, capturing the influence of a layer in grouping all nodes (this overlap centrality is named layer *entanglement* [61]). In terms of search results, this entanglement would capture this influence of semantic concepts over the result space.

A large number of layers can also be turned into an advantage by forming a secondary network [60], the *Layer Interaction Network* (hereafter *LIN*). The *LIN* models the overlap of layers on edges and offers interesting opportunity in terms of interaction and coordination [60].

1.2 Contributions

This paper extends our first introduction of the *visual cloud* [58], which presents a tag-cloud-inspired visualization algorithm that includes images and heat-map, built using a multilayer network. While the original paper focuses solely on the construction of the visual cloud, this paper contributes with a full search system built on top of multilayer networks, and uses the visual cloud as a support for navigation (as illustrated in Figure 1):

- We propose the design of a search visual analytics system based on Brehmer *et al.*'s search tasks recommendation [11].
- The whole search engine is driven by a single data abstraction that are multilayer networks as formulated by Kivelä [32]. The multilayer network captures both visual and textual index, while driving all its visual and analysis elements.
- The system offers users to formulate complex Boolean queries based on a triplet formalization.

- Extracting hierarchies of layers and using them to draw the visual cloud, enriched with layer analysis were already contributions of our original paper.
- This paper refines our visual encoding and additionally defines multilayer network operations to support search with a series of linked-highlighting and leapfrog interactions.
- We apply our system to a new dataset from the Internet Movie Database (IMDB[1]) to show its generic character.
- We introduce finally new evaluations and use cases.

The paper is written following the recommendations of Munzner [50] (properly separating *tasks*, *abstraction*, *visualization*, *interaction*, and *algorithms*) as follows. After reviewing the literature in the next section, we set up our design requirements in Section 3. We describe our data and preprocessing in Section 4. We then explain how to model our search results with the multiplex abstraction in Section 5. Analysis based on this abstraction is then exposed in Section 6. This leads us to introduce our visual encodings in Section 7 and interactions in Section 8. We discuss our results in Section 9 before concluding.

2 Related Work

Search Result Clustering (SRC), is a very popular research area. It proposes to organize the snippets returned by a search engine into meaningful thematic groups so users could look at search results before exploring the clusters themselves and target their own search [31]. SRC is often described as a four step process [14]: *results acquisition*, *preprocessing*, *clustering*, *labeling*, and *visualization*.

Although our work is not focused on the clustering of search results itself, search engines providing clustering often need to visually provide their results [63] [20] [80] [35], adding features to the standard search interface to represent meaningful topics. Furthermore, most of them focused at best onto cognitively costing file-hierarchy type of visualization.

KartOO [34] is a meta-search engine that proposed visualization of results based on a thematic map. The map is basically formed from the semantics links between results, with a high information overload cost that would limit possible results. Grouper [63] is a historical document clustering interface for meta-search engine. The clusters are displayed in a very primitive way: results are presented in a table, each row referred to as the summary of cluster. SnakeT [20], WICE [80], and Vivisimo [35] each present their own way to cluster results and build a hierarchy presented as a file-tree corresponding to (sometimes overlapping) lists of results. Carrot[2] [55] is an active project that provides interesting visualizations of their clustering: in addition to the popular file-tree view, they provide a multi-level pie chart (entitled “Circles”), and a Voronoi tree-map (entitled “FoamTree”). Unfortunately, even after multiple queries trials we only obtained a flat clustering, ruling out hierarchical representations. In addition, none of the works above combines multimedia information, neither have they approached the word cloud, and even less a visual cloud as an option to explore cluster

[1] see: `https://www.imdb.com/`

content. Interaction design (coordinated brushing and linked highlighting) has not been much proposed either. We should also salute the effort of Google Image Search to add tags of relevance above the image results to help complete the image search (see Fig. 2). In turn, we propose a visual analytics system that is designed to support all different search tasks [11] in a systematic way. It includes the task of *exploration* that our system supports by augmenting the regular search interface with overviews and coordinated interactions. To this end, we design a visual cloud that is adapted to the search query, that combines multimedia information through an overview showing images and text.

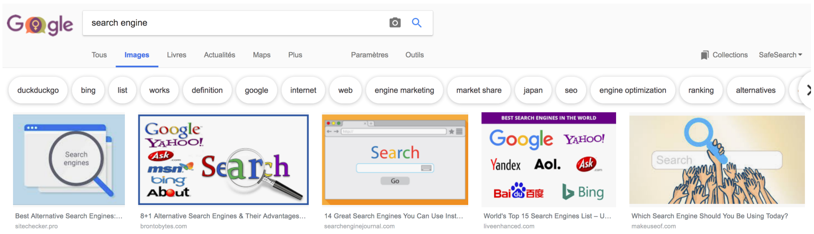


Fig. 2 A Google search image query for *"search engine"* enriched with tags (*captured on 2019/03/08*).

Beyond traditional topic modeling [47], graph based methods [62] [19] have been proposed to explore groups of documents, but mostly without integrating multimedia indexing or visualization. The work of Scaiella *et al.* [62] purely focuses on topic clustering in search results, and creates a hierarchical descending co-clustering of the topic-document graph. Close to this work, the entanglement [61] measures document group cohesion by association of topics based on a multiplex network formulation. None of these works investigate data manipulation as a support of search tasks. In turn, we propose to form hierarchies of layers in a multilayer network formalization, which resembles a hierarchy of topics and optimizes group cohesion. This hierarchy drives our exploration, and defines, together with analysis measure, data manipulations so that any interface interaction corresponds to a multilayer network operation.

In turn, visualization addresses the display of search engine results by trying to end with the *list-of-snippets* representation and organize results spatially [25], or even by treemaps [16] [54]. While they are good for coordination and contextualization, they don't provide space efficiency. In contrast, we propose an integrated methodology for multimedia data, with exploration, analysis, interaction and search refinement. Note that multilayer graphs have also been used to investigate groups of documents [60], but never in the context of a search engine.

Tag clouds provide a good overview of the semantic space [66] and tag placement has been improved to convey semantic proximity [8] [77] [78]. However, none of these approaches are adapted to heterogeneous data or multiplex networks, which would enable coordinated interactions [60]. We used the fastest tag cloud method, Wordle [71], for its capacity to be easily constrained and coordinated.

Although a bit further from our focus, visual analytics proposing to explore topical analysis of news events is also relevant. Krstajić et al. [37] have proposed a unique way to follow evolution of news topics over time inspired by Sankey diagrams. News Rover [41] combines multiple sources across many media hyperlinked through image and topic matching: the visual analytics consists of a force-directed neighborhood map of news documents for exploration, and a second visualization puts a topic in context of time and subtopics, but without semantic overview as we propose it. Network of Names [33], allows users to create narrative graphs from a source of documents by semi-automatically tagging social relations in a graph derived from news. OTMedia [27] explores the relationship across different sources of information in an information retrieval system. The NHK News 7 dataset has also been tackled by diverse works: one proposed linked videos [42], another, 3D timelines of clustered data [30]; summary news videos are generated in [28], and political networks explored in [59]. We combined preprocessing of the last two works, while proposing a novel semantic analysis and interactions. The tasks of these works mostly differ from ours, being very domain specific. In turn, we address the more generic issue of contextualizing a group of query's results, designed in support of search tasks, with interaction and overview as pivot elements of our visual analytics.

3 Design Requirements

Our system is designed to support search tasks in multimedia databases. Following Brehmer *et al.* [11], tasks are described by asking three questions that serve to disambiguate the means and ends of a task: *why* the task is performed, *how* the task is performed, and *what* are the tasks inputs and outputs.

Similarly, context is often investigated under the *Five Ws* [67]. Based on our data (further described in Sec. 4) time (*When?*) is atomic and attached to a data instance itself (such as release date, e.g.); people (*Who?*) are extracted or provided separately, from pictures or metadata, and sometimes with roles (actor or director). Keywords mainly provide content for *What?*, although they may provide information answering *Where?*, when relevant. They may also answer *Who?*, and may even give additional reference to contextual time marks *When?* (as opposed to time marks attached to the document itself such as its creation date).

When considering search tasks, Brehmer distinguishes four cases depending whether the user is looking for a *known/unknown* target, and whether the user knows *where* to look for. In our system, a target would translate as a multimedia element or a specific element (person, date, actor, *etc.*), and a location would correspond to the context of an element (composed as a combination of date, topic, people, *etc.*).

As a consequence, our design needs to support the following four tasks:

- **T1**: *Look-up*. The search target and search location are both known by the user, such as finding one specific movie.
- **T2**: *Locate*. The search target is known but not its location, such as verifying the cast of a movie.
- **T3**: *Browse*. The search target is not known but its location is known, such as finding movies from a specific actor during a specific time frame.

– **T4**: *Explore*. Neither the search target nor its location are known, the user is rather searching here for characteristics, such as the most influential actors during a certain period of time.

Supporting **T4** requires to give a clear visual representation of both the results of a search query and its context. Exploration is done through an iterative process that can be captured through the Visual Information-Seeking Mantra: *"Overview first, zoom and filter, then details-on-demand"* [65]. This mantra helps us define three sub-task requirement:

– **T4.1**: *Overview first*. Our system should provide an overview of the context formed by search results.
– **T4.2**: *Zoom and filter*. Users should be able to subset search results based on contextual criteria.
– **T4.3**: *Details-on-demand*. Users must be able to access to individual search result.

In order to support **T1**, **T2**, and **T3**, we propose to implement all capabilities of a search engine, indexing multimedia documents by their semantic content, with Boolean and criterion-specific search [68] [70] to enhance search in known locations (**T2**, **T3**). Access to search results covers the sub-task **T4.3**.

In support of the sub-task **T4.1**, we propose to use a *tag-cloud*-inspired visualization of the search result context that will present overview of mainly *Who?* and the larger *What?*, and to use a time-bar chart that will give overview over time (See Section 7 for detailed visual encoding). These come along with interaction, necessary for sub-task **T4.2** (described in Section 8). Following Munzner's recommendations [50], we propose the use of one common data abstraction that a *multilayer network* in support of all sub-tasks of **T4**.

4 Data and Preprocessing

Providing a clear task description is but a first step. Tasks then need to be mapped to data and operations performed on these data when carrying tasks. This section takes a closer look at the data we consider. We describe two multimedia document databases, one for each of the use cases we consider.

A first database is built from Japanese news archive and we build a second dataset from a subset of the Internet Movie DataBase IMDB5000[2]. As we shall see, these two use cases are captured in a single unifying framework leading to a generic approach when querying the data (see Section 4.3).

4.1 NHK News 7

Our data consists in a 12-year archive of daily Japanese public broadcast NHK News 7 [59], from 2001 until 2013. Each program is 30 minutes long with synchronized closed captions, about 6 months of 24/7 viewing in total. A program is composed

[2] Available from Kaggle at `kaggle.com/carolzhangdc/imdb-5000-movie-dataset`

of different *news segments*. As provided by Ide [29], we obtain the segments using a sliding window of topic distribution (Fig. 3, top). The extracted topics being very noisy, we further extract textual semantic information, for each segment, with a keyword extractor trained for news documents (including named entities) [21]. The Japanese keywords are translated to English with Bing Translator[3].

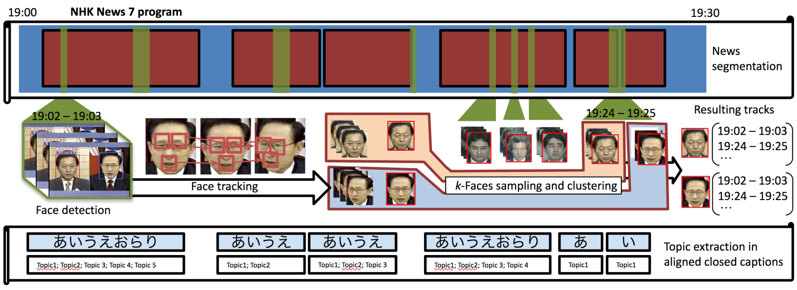


Fig. 3 Indexing each video segment: closed captions-based segmentation, face tracking, and keyword extraction.

We use the face detection and tracking proposed in [59] (Fig. 3). Faces instances are detected in each frame [72], then regrouped with point tracking [64] creating *face-tracks*, sampled with k-faces [52], and represented using the average of its 128-dim OpenFace embedding vectors [3]. Face-tracks are clustered using GreedyRSC [39]. About 3,000 clusters are manually annotated, resulting in over 15,000 face-tracks of 139 public figures. We index each video segment with date-time of broadcast, keywords, and face-tracks. A query can then be specified upon any of these criterion, returning a subset of news segments, including their associated semantic concepts, *i.e.* keywords and detected faces.

4.2 IMDB 5000

IMDB5000 comprises 5,044 movies that have been chosen based on their revenue or critic reviews. Movies span from 1925 to 2016. Each movie is provided with metadata including title, director (2,399 different directors) and genre (26 genres in total). Because the dataset propose only limited cast and keywords (up to 3 per movie), we scraped additional metadata directly from the IMDB website. For each movie, we gathered: story lines, the full cast, actor and director pictures, and keywords indexing the movie scenario. In total, we crawled 37,538 unique actors (having 24,767 face pictures), and 71,781 unique keywords. Collecting such rich metadata allows to query the dataset in a variety of ways. A query will return a group of relevant movies equipped with their metadata (title, genre, cast, actor/director pictures, keywords).

[3] See `microsoft.com/translator`

4.3 Querying Datasets

Any of the considered task **T1** through **T4** requires to initially form a query. To support search tasks where the target or location is known (**T1** and **T3**), we provide a search query interface. The query results form an informational space organized into a multilayer network (see Section 5.1).

The metadata we consider allow to specify a search query as a triplet (*time-frame*, *person*, and *keyword*) $q = (d, p, k)$. As usual, leaving one of these criteria empty returns "any" available value; for example, $q = (*, p, *)$ returns all documents related to person p.

Search queries may be combined to form more complex queries. Boolean operators $\neg$, $\wedge$,$\vee$ (*not*, *and*, *or*) are manipulated through intuitive actions. Searches on specific combinations of keywords/people over specific periods of time are thus easily performed. For example, Fig. 4 illustrates a search for all *Spielberg*'s movies on *war* and *nazi* that are not *Indiana Jones* (*i.e. Schindler's List* and *Saving Private Ryan*).

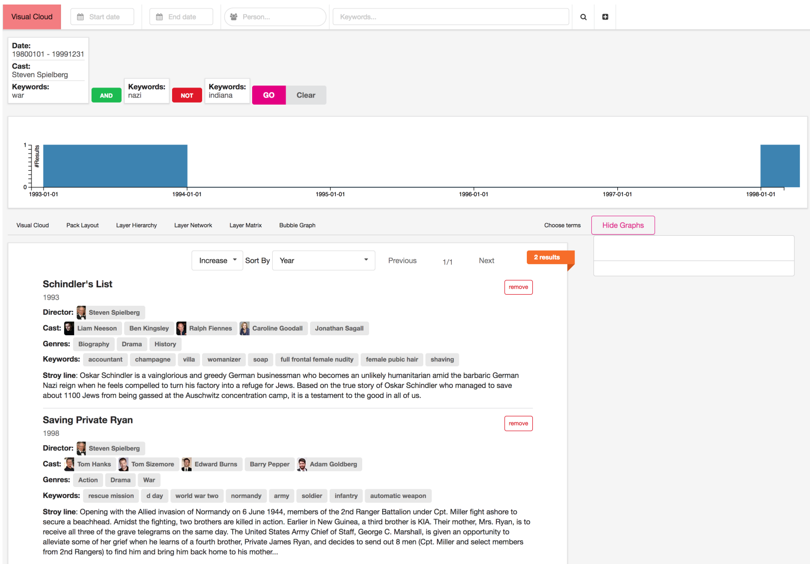
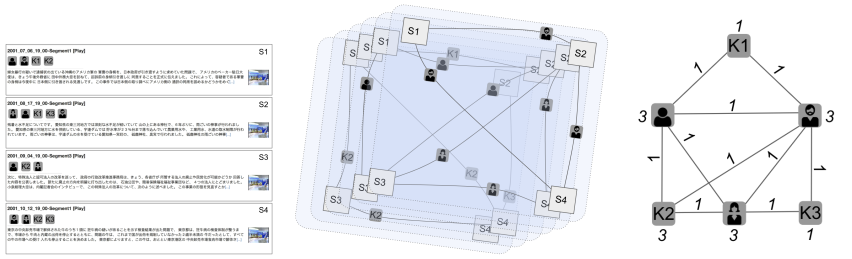


Fig. 4 Searching for *Spielberg*'s movies in the *80's* that are about *nazi* and *war* but not on *Indiana Jones* bring forward the *Schindler's List* and *Saving Private Ryan*.

5 Multilayer Network as a Data Abstraction

Each query returns a certain number of documents within a user-defined maximum (news excerpts or movies), each indexed with a set of metadata: time related information, persons, keywords (as illustrated in Fig. 5.a).

As we shall see, a graph can be built based on these elements allowing to leverage its topology for further analysis in different ways, clustering or ranking of various



(a) (b) (c)

Fig. 5 Model and data abstraction of the search results: a) query results with their index, b) results of (a) as a multilayer network, c) the Layer Interaction Network (the weighted *LIN*) associated to (b).

graph elements [79] [45] [62] [73] [61]. In order to faithfully encode all metadata, we build a so-called *multilayer graph* [32] [73] [17] [61] and use it as the query context.

Multilayer networks have recently been formally introduced by Kivelä *et al.* [32], although one can trace them back to earlier literature in various application domains (see [24]). A multilayer network is built on top of nodes, just as a usual network (graph) is. In our case, nodes are documents (news excerpts, movies) whose set is denoted as V. Edges, as usual, connect nodes $u, v \in V$. A fundamental concept here is that of a *layer*: nodes and edges spread over different layers according to their "types"; a multilayer network is thus a "network of networks" [22].

In our case, layers L correspond to metadata items indexing and/or extracted from documents, such as persons and keywords, for instance. A node $v \in V$ thus appears on a set of layers; on the contrary, an edge appears on a single layer encapsulating the "type" the edge bears. Obviously, we could optionally consider a property-based graph and simply map metadata to edge attributes. The notion of layers however brings in added value as we shall see (see Section 5.1).

A multilayer network formally corresponds to a quadruple $M = (V_M, E_M, V, \mathbf{L})$ where V_M (with subscript M for "multilayer") properly describes nodes together with layers on which they appear (and is formally a subset of $V \times \mathbf{L}$, for more details, the reader is referred to [32]); edges $e \in E_M$ connect nodes in V_M. The set V references the original set of "ordinary" nodes while $\mathbf{L} = (L_a)_{a=1}^{d}$ stores subsets of layers (layers can be combined; again, see [32]). From now on, we shall denote the set of (all) layers as L, and individual layers as $l, l', l'', \ldots$ Recall that individual layers map to metadata items (such as keywords or persons). We shall also refer to metadata items as (semantic) "concepts", hence interchangeably using the terms *layer* or concept as synonyms.

Denote by T the set of metadata indexing documents. As mentioned above, each layer L is induced from a metadata item $t \in T$. For sake of simplicity, we denote an edge as $e = ((v_1, t), (v_2, t)) \in E_M$. Such an edge exists if two documents (resulting from a search) are annotated with the same metadata $t \in T$. We also consider time and specify broadcast or release dates, thus extending nodes in V_M as triples *node* $\times$ *layer* $\times$ *date*.

Although the resulting multilayer network is very difficult to visualize *as is* (see Fig. 6 for a real example), using this model as a data abstraction [50] allows us to manipulate the result space together with its context all at once.

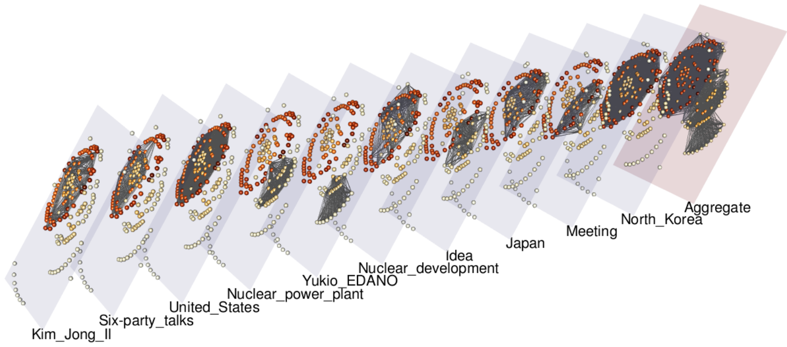


Fig. 6 An example multilayer model of a result space visualized with MuxViz [17]. We queried the term *nuclear* from the NHK dataset, showing 140 results and 192 layers in total. For sake of legibility, we only visualize here the top 10 layers in blue, and the 'aggregated' overview layer is in red.

5.1 The Layer Interaction Network (*LIN*)

Now that we have defined query-derived multilayer networks, we will introduce how using this data abstraction may support tasks **T4.1** (overview) and **T4.2** (filter/zoom).

The purpose of an overview [65] is to give in a glimpse the idea of a whole collection (for example, in a country map, an overview gives user the current zoom level with relation to the country). When it comes to multilayer networks, the overview could be an agglomerated view, as proposed in MuxViz [17], where several layers are combined into a single network. Because our multilayer network is a proxy to model search results, the agglomerated network often forms a clique, and because the number of layers quickly rises over the dozen, the pure multilayer view rapidly becomes unpractical as illustrated in Fig. 6 (edge cluttering becomes too important [23]).

Past research has demonstrated the interest of using a graph of topics to cluster search engine results, as argued in [62] *"a graph of topics provides a contextualization of snippets."* We extend this idea and form a Layer Interaction Network (*LIN*) of multilayer network (seen as a generalization of a "graph of topics"), borrowing from [61] [60] where the *LIN* helps navigate groups of documents.

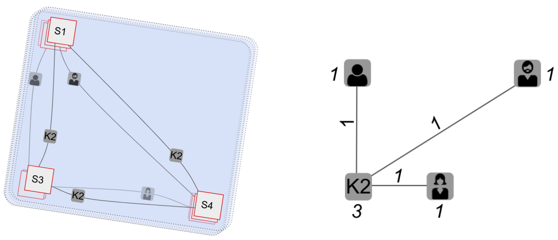
The *LIN* can be seen as a layer co-occurrence graph. Nodes of the *LIN* correspond to layers $l, l', l'', \ldots$ in the multilayer network M. An edge f between layers l, l' exists whenever these two layers overlap; that is, when there are nodes $u, v \in V_M$ that are connected in the multilayer network through both layers l and l', with edges $u \rightarrow_l v$, $u \rightarrow_{l'} v$ on layers l and l'. The weight of the edge $f = (l, l')$, denoted as $n_{l,l'}$ equals the number of times layers l, l' co-occur – that is, the number of node pairs $u, v \in V_M$ that are connected on both layers. By extension, $n_{l,l}$ is the number of edges on layer l.

Fig. 5(c) shows a toy example of a *LIN* formed of six layer nodes, three persons and three keywords K_1, K_2, K_3. The edge between keyword K_1 and a person p (either left or right) corresponds to documents d_1, d_2 that are both connected through keyword K_1 and person p. There is a unique such pair of documents, leading to a weight of 1 for the edge.

5.2 Data Manipulations

We propose to filter and zoom on elements of interest so that we may refine the search space. This not only support exploration **T4**, but can actually help also in support of browse task **T2** when the search target is not fully specific. Any of these operations will subset our initial multilayer graph M to a smaller multilayer graph $M' \subseteq M$ (less nodes, less layers).

Filter and zoom tasks [65] are implemented as user interactions on the LIN affording to focus on items of interest. Layers here simultaneously act as a main concept when querying and interpreting data, and as a driver to perform tasks. As we shall see, most tasks require to compare layers, examine whether documents spread on numerous layers, etc.

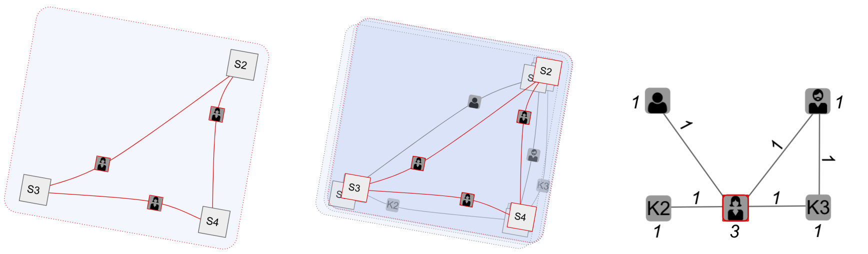


(a) (b)

Fig. 7 Example of temporal selection, the induced subgraph from a subset of nodes (in red) in the multilayer network (a) from our toy example, with (b) its corresponding LIN.

The temporal aspect is straightforward, and a selection of a timeframe (d_0, d_1), restrains all multilayer nodes $v_{M'} = (v, l, d) \in V_{M'} \subseteq V_M, v \in V$ such as $d_0 \leq d \leq d_1$. From this subset of nodes, we may define an *induced multilayer subgraph* M' on a subset of nodes $V' \subset V$ and subset of layers $L' \subset L$ relevant to the time frame. Note that the resulting multilayer subgraph M', layers may have entanglement that differ very much from theirs of the original M (both qualitatively and quantitatively, see section 6.1).

Layers can be used to drive interaction; selecting, adding or removing layers can be seen as a form of semantic zooming and filtering. In terms of interaction, the relative entanglement of layers may be updated as the considered subgraph evolves (illustrated in 7).



(a) (b) (c)

Fig. 8 (a) Selecting one layer (in red) from our toy example corresponds to select its edges. (b) *Leapfrogging* from this selection associate the other layer overlapped by these layers in the original multilayer network, c) the associated LIN to the leapfrog (with selected layer in red).

The notion of layers allows a quite natural and most useful interaction introduced in Detangler [60] as a *leapfrog* interaction. Assume a subset of layers $L' \subset L$ has been selected from M (over all layers L). Leapfrogging from L' to $L'' \supset L'$ can be performed by *expanding* L' to any other layers $\bar{L}$ on which a pair of layers $l, l' \in L'$ also co-occur. Conversely, we can restrict layers to $L''' \subset L'$ on which *all* pairs of layers l, l' appear.

This is illustrated in Fig. 8. Again, the relative influence of layers within L'' may strongly differ from their original influence in L or L'. With layers as semantic concepts, this helps to investigate how influential are a group of concepts in their associated results, but also *which other concepts* may be influential (or not) on these same results. For example, from a query of movies directed by *Martin Scorsese*, this can help investigating the most influential actors/concepts of all movies involving *Leonardo DiCaprio*.

6 Analyzing a Multilayer Network

To help support overview **T4.1** and guide zoom/filtering **T4.2**, we need to guide users through a set of indicators that may be embedded in the visualization. Since we do not have causality or frequency analysis as a requirement, we will stick to occurrence analysis in the ordered 1-dimensional variable that makes the time aspect. The semantic concepts aspect requires more sophisticated cues.

Depending on the query: a group of concepts (that is, a group of nodes in the *LIN*) may form together a topic of interest. The role of a concept may greatly vary whether central or satellite to the query, context may give sense to semantically ambiguous/hollow concepts. Satellite concepts may guide the search location for browsing task **T3**. These questions may be addressed by observing the relationships between layers, and how they overlap using our data abstraction.

Furthermore, hierarchical relationships can provide a meaningful navigational mechanism by organizing information into a small number of hierarchical clusters [81] [4]. Detangler [60] has reported that the *LIN* captures the semantic context with *"a sense of hierarchy"* [60]; however, the *LIN* drawn "as is" tends to quickly become a fur-ball as the number of concepts increases. In this section, we address this problem and propose an algorithm that extracts hierarchy by maximizing layer overlapping.

6.1 Layer Entanglement

The multilayer network data abstraction provides a mean of measuring the "influence" of a layer in the whole multilayer network. We rely on a layer entanglement measure defined in [60] to compare layers and evaluate the internal cohesion of a group of documents (where cohesion gets higher as documents are linked to multiple layers).

The *layer entanglement* γ_l for each layer l is recursively expressed: the entanglement measure of a layer feeds on the entanglement measure of the layers it is entangled with. Computationally speaking, this translates into a recursive equation: $\gamma_l.\lambda = \sum_{l' \in T} \frac{n_{ll'}}{n_l} \gamma'_l$, following a pattern similar to that of pagerank or eigenvalue centrality (see [75]).

The vector $\boldsymbol{\gamma}$ of all layer entanglement is the right eigenvector corresponding to maximum eigenvalue λ of the layer overlap frequency matrix $C = (c_{ll'})$ where $c_{ll'} = \frac{n_{ll'}}{n_{ll}}$ (as described in [61]).

Layer entanglement has proven to be more reliable (when interpreting results into data context) than using plain co-occurrence, for instance (see [57]). Entanglement is a more sophisticated measure highlighting the relative importance of semantic concepts.

6.2 Overlap Similarity and Dissimilarity

Our original paper [58] extended the work of Renoust *et al.* [61] by introducing a notion of similarity between layers and groups of layer: the *Layer Overlap Possibility* (LOP). We say that layers l and l' *overlap* exactly when $n_{l,l'} \neq 0$. The LOP in a sense measures how much two layers overlap. When $n_{ll'} = 0$, we say that layers L, L' are *disconnected* (or disjoint).

We define the LOP of two node layers $l \in L$ and $l' \in L$ connected in LIN as follows:

$$LOP\,(l, l') = \frac{n_{ll'}}{n_{l'l'}} \cdot \frac{n_{ll'}}{n_{ll}} \in [0, 1] \tag{1}$$

when layers l, l' overlap, and

$$LOP\,(l, l') = -\frac{n_{ll} + n_{l'l'}}{|E|} \in [-1, 0] \tag{2}$$

when layers l, l' are disconnected.

The value $LOP(l, l')$ is maximized ($LOP\,(l, l') = 1$) if and only if two layers exactly coincide in M ($E_l = E_{l'}$).

With $|E|$ the number of connected pairs of nodes in the multilayer network M. The LOP value is then negative, with minimal value $LOP\,(l, l') = -1$ when the two layers l and l' cover all connected nodes in M but do not overlap.

Following the same idea, the LOP between a layer l and a group of layers S is formulated as:

$$LOP\left(l,S\right) = \sum_{l' \in S} LOP\left(l,l'\right),\ l \neq l' \tag{3}$$

Finally, the LOP between two groups of layers S and S' ($S \neq S'$) is then defined as:

$$LOP\left(S,S'\right) = \sum_{l \in S} LOP\left(l,S'\right) \tag{4}$$

We may verify that $LOP\left(S,S'\right) = LOP\left(S',S\right)$. This formulation of similarity could be seen as analogous to forces in a directed layout algorithm [53]. Overlapping layers should tend to "attract" each other, while disconnected layers should "repulse" one another, placing them far apart. If we use LOP to group layers together, high negative values avoid unconnected layers to be assigned to the same group.

6.3 Layer Hierarchy

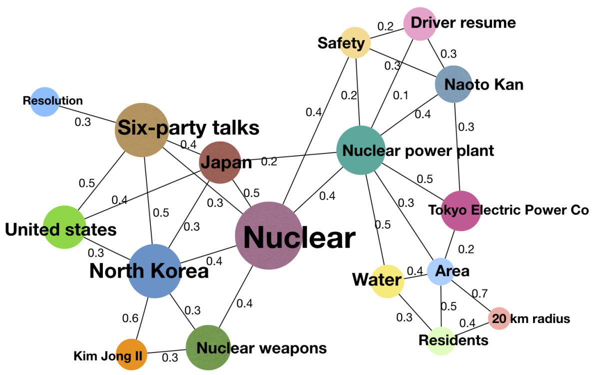


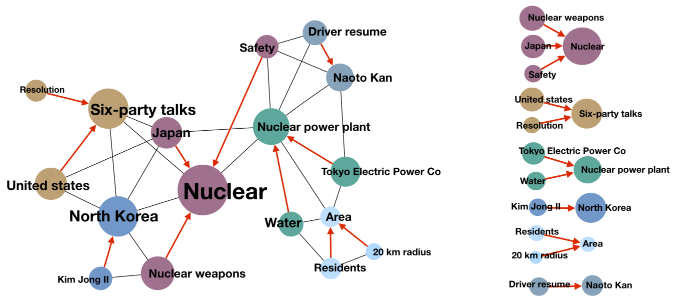
Fig. 9 A LIN associated with a search on *'nuclear'*, node size correspond to layer entanglement, $LOP(l,l')$ is registered on the edges. At the initialization, each node is associated to a unique label (node color)

To extract a hierarchy of layers, our strategy is to identify layers and nodes with optimal cohesion (entanglement homogeneity). Since it is an NP-hard combinatorial optimization problem, Renoust *et al.* [61] did not offer any solution. Keeping in mind that we have a multilayer subgraph corresponding to each (group of) layer(s), we propose a heuristic solution to maximize entanglement homogeneity in groups of layers (*i.e.* in a subgraph induced by a leapfrogged group of layers). This algorithm aims at building trees of which nodes correspond to layers, by aggregating them through navigation of the *LIN*. We follow the greedy heuristic introduced in the *Louvain* method for optimizing modularity [10].

The algorithm runs in three steps (Figs. 9-12). The first step aims at initializing parent-children links. The second step associates all children nodes to their closest layer (or group of layers). The last step iteratively assembles groups of layers.

To initialize the algorithm, each layer is first associated (*labeled*) to its own group $\mathcal{L}(l)$ (Fig. 9). Layers are aggregated in the process of the algorithm through the topology of LIN such as two layers l_i, l_j will be connected if they are linked by an edge in LIN (*i.e.* when they are overlapping in M). Orientation of the association (l_i, l_j) is decided from $max(\gamma_{l_i}, \gamma_{l_j})$ (rooting the hierarchy on the most entangled layer).

6.3.1 Step 1: Generate "parent-children" pairs for all layers



(a) (b)

Fig. 10 Step 1, creating *parent-children* associations from the LIN by maximizing layer-layer LOP (a). The resulting trees in (b). The hierarchical edges are highlighted in red.

In this step (Fig. 10), each layer l_i, that is not a parent already, will be associated to a layer group $\mathcal{L}(l_j)$ in its neighborhood $l_j \in \mathcal{N}_{LIN}(l_i)$ such as $LOP(l_i, l_j)$ is maximal. Each layer is then labeled with its parent's group $\mathcal{L}(l_p)$. All the pairs form now $\Lambda = \{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_k\}$ groups of layers.

Algorithm 1 Generate *"parent-children"* pairs for all layers

procedure INITIALIZATION
  for l in L **do**
    assign to l a unique label $\mathcal{L}(\updownarrow)$
procedure STEP 1
  for $l_i \in L$ **do**
    if l_i is not *parent* **then**
      $l_j \leftarrow n \;\;|\;\; \max(LOP(l,n)), \;\; n \in \mathcal{N}(l)$
      if l_j is not *parent* **then**
        $l_x \leftarrow parent \;\;|\;\; \max(\gamma_{l_x}), \;\; l_x \in \{l_i, l_j\}$
      $\mathcal{L}(l_y) \leftarrow \mathcal{L}(l_x) \;\;|\;\; l_x$ is *parent* of $l_y, \; l_x, l_y \in \{l_i, l_j\}$

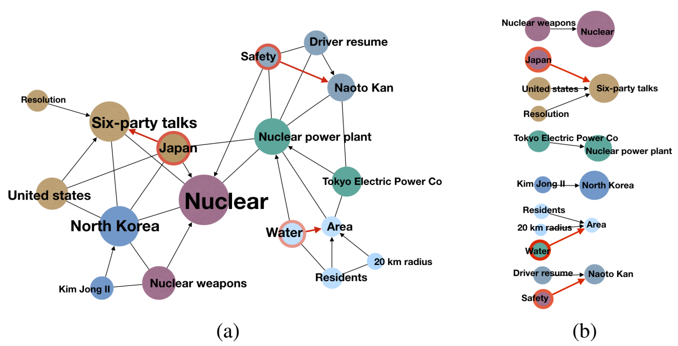


Fig. 11 Step 2, moving layer nodes to their closest neighbor (in terms of maximum *layer-layer* or *layer-group* LOP) from the LIN (a). The resulting trees in (b). The moved layer nodes are highlighted in red.

6.3.2 Step 2: Link children to their closest layer or group of layers

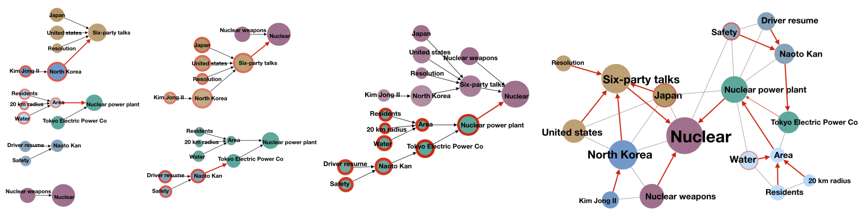
We need now to associate layers to the group in which they can maximize entanglement. We now compare the group $\mathcal{L} \in \Lambda$ in which a layer l_i maximizes $LOP(l_i, \mathcal{L})$ to the neighboring child layers l_j that also maximizes $LOP(l_i, l_j)$. When a pair of layers maximizes entanglement together, as opposed to be associated to a group of layers, we create a new label for both layers. This step avoids too coarse aggregations often induced by very overlapping layers (sometimes corresponding to a queried concept). Because changing a layer's group may impact other layers in a same group, we continue updating associations until no change occur anymore. Results are illustrated by in Fig. 11.

Algorithm 2 Link *children* layers to their closest layer or group of layers

---

procedure STEP 2
  $\Lambda = \{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_k\}$ (groups formed from *Step 1*)
  repeat
    $changed \leftarrow False$
    for $l_i \in L \;\; | \;\; l_i$ is not *parent* **do**
      $\mathcal{L}_j \;\; | \;\; \max(LOP(l_i, \mathcal{L}_j)), \;\; \mathcal{L}_j \in \Lambda$
      $l_k \;\; | \;\; \max(LOP(l_i, l_k)), \;\; l_k \in \mathcal{N}(l_i)$ and l_k is not *parent*
      if $LOP(l_i, \mathcal{L}_j) > LOP(l_i, l_k)$ **then**
        if $\mathcal{L}(l_i) \neq \mathcal{L}_j$ **then**
          $\mathcal{L}(l_i) \leftarrow \mathcal{L}_j$
          $changed \leftarrow True$
      else
        $l_x \leftarrow$*parent* $\;\; | \;\; \gamma_{l_x} = \max(\gamma_{l_a}, \gamma_{l_b}), \;\; l_x \in \{l_a, l_b\}$
        $\mathcal{L}(l_x) \leftarrow$ new label
        $l_j \leftarrow l_i \leftarrow \mathcal{L}(l_x)$
        $changed \leftarrow True$
  until $\neg changed$

6.3.3 Step 3: Regroup all groups in a forest



(a) (b) (c) (d)

Fig. 12 Step 3, grouping groups of layers to their closest neighbor (in terms of maximum *layer-group* LOP) in 3 steps (a-c) until only one group is obtained (c). The moved layer nodes are highlighted in red. The hierarchy overlaying LIN in (d) shows different to what may be intuited (especially the link between *Kan* and *TEPCO* on the right), but we should bare in mind that this hierarchy optimizes entanglement homogeneity in the induced multilayer subgraph.

The last step computes a hierarchy between groups, by simply comparing LOP of neighboring groups together. We start with the pair of neighboring groups displaying the highest LOP in the graph (Fig. 12). Hierarchical link orientation depends on largest layer entanglement of the parent layers of each group. We stop grouping when there are only negative distances (*i.e.* $LOP(\mathcal{L}_i, \mathcal{L}_j) \leq 0,\ \forall i, j$) or if we obtain $|\Lambda| = q$ groups (by default $q = 1$, but it may be user specified). Note that we may obtain multiple hierarchical trees, and the most obvious case is when LIN is not always only one connected component.

Algorithm 3 Regroup all groups in a forest

procedure STEP 3
  $\Lambda = \{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_k\}$ (groups formed from *Step 2*)
  repeat
    $\mathcal{L}_x, \mathcal{L}_y \quad | \quad LOP(\mathcal{L}_x, \mathcal{L}_y) = \max(LOP(\mathcal{L}_i, \mathcal{L}_j)),\ \ \forall(\mathcal{L}_i, \mathcal{L}_j)$
    $u \leftarrow max(\gamma_{Parent(\mathcal{L}_k)}),\ \ k \in \{x, y\}$
    $v \leftarrow min(\gamma_{Parent(\mathcal{L}_k)}),\ \ k \in \{x, y\}$
    $\mathcal{L}_v \leftarrow \mathcal{L}_u$
    $Parent(\mathcal{L}_v) \leftarrow Parent(\mathcal{L}_u)$
  until $|\Lambda| = 1$ or $LOP(\mathcal{L}_x, \mathcal{L}_y)) \leq 0$

7 Visualization Elements

Now that we have defined all data models, analysis and manipulation, we may introduce our design choices for the implementation of the visual analytics system. Our system revolves around three main elements: a search bar, an overview (of two aspects), and a list of results (see Fig. 1).

7.1 Search Bar

The search bar is the user's entry to the system. It proposes to search among 3 criteria corresponding to *When?*, *Who?*, and *What?* following now standard methods: dates are proposed through a date picker, people are proposed with a suggestion list from the annotated people, and keywords are free-typed.

Each search query forms a triplet *(Timeframe, Person, Keyword)* with a default AND association between them (empty criteria meaning all elements). We provide visual assistance to associate multiple triplets without limit to form complex boolean queries with all operators (*not*, *and*, *or*) as illustrated in Figure 1(a).

Timeframe checks the resulting dates, *Person* checks the identified people, *Keyword* is free indexing. All multimedia documents are fully indexed, not only from their content description, but all textual metadata. By having *Keyword* as a free form, users can, for example, search movies by their titles (giving access to items for design task **T1** *Look-up* and **T2** *Locate*). Using a combination of criteria, users can specify the "location" (in terms of date/person/keyword) in support of *Browse* (task **T3**).

7.2 Search Results

Not unlike most traditional search engines, search results are displayed under the form of a list of snippets, so that users may instantaneously recognize their result for location-known search tasks (*Look-up* **T1** and *Locate* **T2**). Each result displays a title, a textual content (*i.e.* storyline or closed captions), in addition to a date and semantic concepts (so the *location* of a result may be instantaneously recognized in support of *Browse* task **T3**).

The search results provides a series of simple but essential interactions. The default ranking is the relevance of a result that we arbitrarily set to be the multilayer centrality of the result by default. This ranking can be changed to any other ranking provided in the data, most importantly the date of results. In support of *details-on-demand* (**T4.3**), search results give access to all detailed information of a document on a click, launching a video, or previewing and linking to the corresponding IMDB webpage (see Fig. 1(d)).

7.3 Overview

The overview is at the heart of our most important contribution, and it is designed to support location-unknown tasks *Browse* (**T3**) and most particularly *Explore* (**T4**). The overview concerns two aspects of the multilayer network, *time* and *semantic concepts* (as illustrated in Figure 1(b) and (c)).

7.3.1 Time Overview

For the time overview representation, we opted for a traditional timeline representation following systematic recommendations [1]: our representation of time is event-based (unit of duration is the same for all elements) and linear, it is a piece given data, univariate, and abstract, and we wish for a static 2D representation. Each result is located on the timeline in a bar chart manner, so user can identify area in time for high/low density of results for the $y-$axis value. The timeline may be enriched with background time-frames when meaningful, like in the case of the NHK dataset (each time-frame corresponding to a different Prime Minister period) (see Fig. 1(b)).

7.3.2 Semantic Overview

The goal of the semantic overview is to serve as a topical map that helps understand both the people and how they are associated in underlying groups of results. Graph of topics have been used in the past [62] [60] but they are very inefficient in terms of space usage, and become quickly cluttered with edges [23]. We chose *Tag Clouds* to root our semantic overview, since they have been demonstrated to provide a good overview of the semantic space [66], making them so popular today.

Our original paper [58] extends this approach with *Visual Clouds* in several ways:

- We first build our cloud directly from our multilayer network abstraction, such that word proximity should preserve layer overlap homogeneity, hence cohesion among the corresponding results.
- We extend the clouds from text to text and images so we may treat seamlessly heterogeneous families of concepts (face detections, people, keywords, *etc.*).
- We propose a new visual variable, a concept's background to form a heat-map, and encode a continuous variable.
- We finally enrich it with a series of original interactions introduced in Section 8.

Concepts are grouped from our hierarchy (Sec. 6), the layout should encode this concept proximity. Concepts belonging to a same groups should be placed close together in the layout. The size of a concept should reflect its relative frequency in the multilayer networks (*i.e.* the layer size). The color of a concept should display a concept group using a categorical color scale. Images have a border, which is colored to their layer group, the same way as font color for text. We additionally constrained the size of images as a function of the max font size in order to keep a visual continuity between the two families of elements. Images may optionally display a label on top (*e.g.* a person's name to help identify a person). The heat value (blue to red) of a concept reflects its layer entanglement: how much it mixes with other concepts: higher layer entanglement are assigned to warm colors to attract attention.

7.3.3 Visual Cloud Generation

Following Munzner's recommendations [50], we may now describe our algorithm to build the visual cloud. To best support users, the visual cloud should be compact and aesthetic. Semantics often carry a hierarchical structure [49], this is the one pointed out in one Detangler's user feed-back: [60] *"It is like a word cloud but better because it gives a sense of hierarchy between terms."* Tapping into this potential, we took this comment seriously to expressly map the thematic grouping resulting from the hierarchical structure extracted through layer overlap in our model. Our first step is to constrain our word cloud model by embedding the hierarchical relationships in the resulting layout, then to build a modified word cloud with images, and finally to highlight regions of interest.

In their evaluation, one user feedback on their topic interaction network was stated as follows:

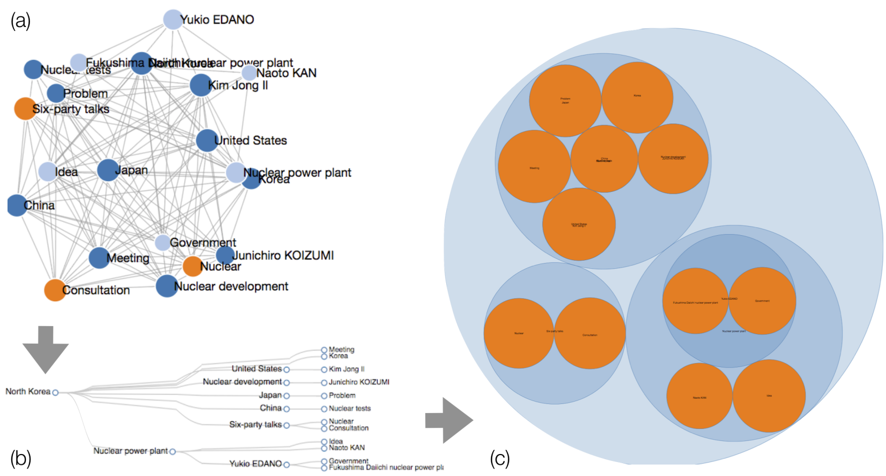


Fig. 13 Initializing the Visual cloud layout: (a) the *LIN* of the query and (b) its extracted hierarchy. We initialize positions with a Pack Layout [74] embedding which clearly do not optimize space but well translate hierarchical separation.

1- Pack Layout initialization: The Pack Layout algorithm [74] uses enclosed diagrams to represent containment (nesting) as the hierarchy (similarly to tree map algorithms). The size of each leaf node reveals a quantitative dimension associated to data points and the enclosing circles show the approximate cumulative size of each subset. Circle packing results in lot of wasted space, while the circle shape is different from the oblong shape of words, or the rectangular bounding boxes of faces. However, it can indicate relative positions of nodes following their hierarchical relationship. Each concept is assigned to the center of the enclosing circle it is represented by (Fig. 13(b,c). Pack Layout creates a leaf-node per node in the tree, instead, we assign parent nodes to the center position of the higher order circles. This results in a more even distribution of position in the plan, giving a relative position for each concept guided by the hierarchy.

2- Visual cloud layout: Wordle algorithm [71] is arguably the fastest tag cloud algorithm. Words initial position can not be strictly specified, and size depends on words relative frequency. Words are introduced one by one to some random position close to the center of the canvas and iteratively placed in the order of frequency. A word is then displaced if it intersects with any previously positioned words. This displacement is made following an increasing Archimedean spiral until no more intersection is found. We use this to our advantage by constraining the spirals with the Pack Layout's circles (center and separation distance), see Fig. 14. We extended the

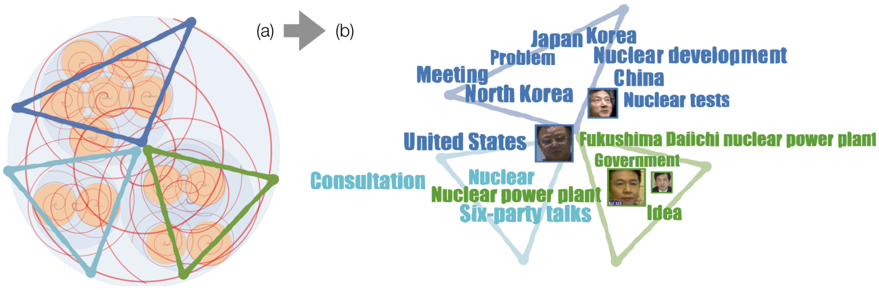
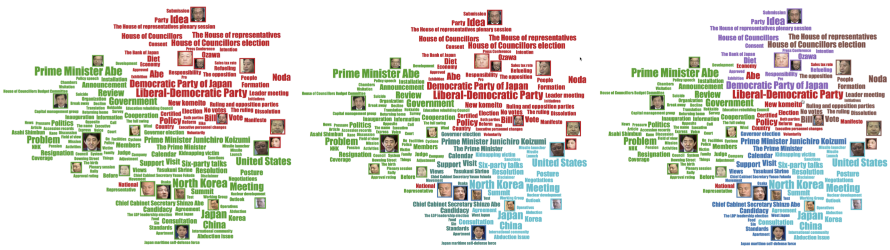


Fig. 14 Drawing the actual Visual Cloud: (a) Positions of words are initialized from the previous step, and rotated around Archimedean spirals [71]. Note how the related position of the three highest levels of the hierarchy is preserved (represented by colored triangles in (a), same color as the visual elements in (b)).

algorithm to take into account any rectangle shape. As a result, concepts are placed in the proximity of their previously calculated position, relatively reflecting their hierarchical structure (as illustrated in Fig. 14).



(a) 2 Groups (b) 4 Groups (c) 6 Groups

Fig. 15 The layout is stable across any number of groups. Note how the layout preserves relative positionning from the hierarchy.

3- Regions of Interest: Textual concepts and image borders are colored upon the group to which they belong and the number of groups can be interactively set. Since it corresponds to a different cut of the hierarchy, it simply updates colors while keeping the layout stable (Fig. 15). The optional background heat-map displays the layer entanglement of concepts so that it contrasts enough with the categorical scale. Words and letters having irregular borders, we paid special attention to maintain their readability. Instead of the heat gradient decaying from the barycenter of a word, the gradient starts from the word's outer bounding box using rectangular rings. We also apply a max blending function in order to increase the saliency of the hottest values (Fig. 16).

The Visual Cloud also provide a large series of rendering parameters, including the number of groups, the maximum number of concepts to display, font, scale and aspect ratio, and alternative views.

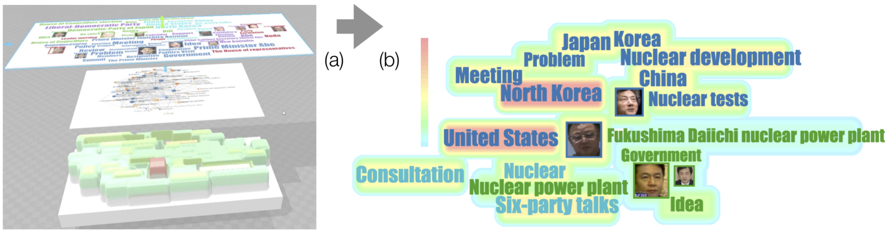


Fig. 16 Since the layout translates the layer hierarchy, the heat map reports properties of the LIN encoding layer entanglement (as schematized in (a)). We display these values in the background of the visual cloud (b).

8 Interactions

The basic search engine interaction covers *details-on-demand* (**T4.3**) on a click on a result. In order to support advanced search functions, we propose a series of zoom/filter interactions (**T4.2**) with linked highlighting (or *brushing-and-linking* [9]) which helps guiding zoom/filters for exploration (**T4**) that is specific to multilayer networks. With zoom and filters, the goal of our interactions are to restrain the subset of results, from either of the two aspects *time* and *concepts*.

8.1 Interacting with Time

Our timeline char is brushable, so that users may draw a time window by dragging the mouse over the timeline. Brushing means dragging this window across the timeline. This defines a time-frame that instantaneously filters all results accordingly.

Following our model and the manipulations described in Section 5.2, this corresponds to a multilayer subgraph M' of our original multilayer graph M, so we recompute our layer measures and report them in the Visual Cloud. To preserve user mental map [48] concept positions are not moved. Layers that do not appear in the filtered subgraph M' are dimmed down in the visual cloud (see Fig. 17). Other concepts are maintained but resized to their corresponding fraction of original size, heat values are also updated accordingly.

Finally, a double-click on the brush opens a new tab with a new query taking into account the selected time-frame $q_o \wedge (tf, *, *)$, where q_o was the original query.

8.2 Interacting with Concepts

Concepts are not brushable but may be selected. Before being selected, concepts may be hovered. Hovering on a concept selects this concept layer in the multilayer network forming a subgraph $M' \subseteq M$, still following our formalism (Sec. 5.2). The induced subgraph subsets the set of original nodes, hence results, so their corresponding dates may be highlighted in the timeline (see Fig. 18 in red).

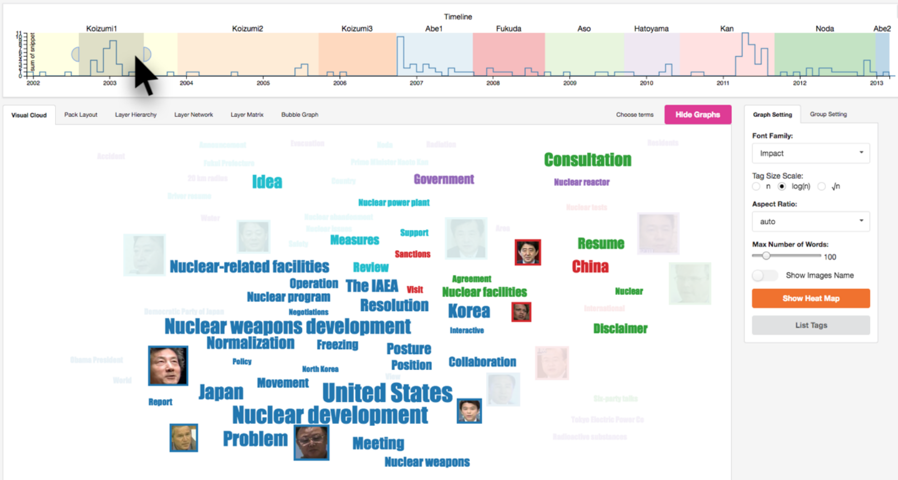


Fig. 17 Selecting a time frame in the timeline highlights the related concepts and recomputes their statistics.

Now we have define the leapfrog manipulation (Sec. 5.2) as a mean to explore other layers overlapping with selected layers M'' in the original graph M. Incidentally this corresponds to the neighborhood of selection in LIN. Hovering a term highlights those other layers by dimming all the non-overlapping concepts. Since building the Visual Cloud implies loosing the topological information of LIN in favor of the hierarchy, we can recover neighborhood information though interaction by hovering over a concept. Because leapfrogging also corresponds to a subgraph, heat values may be recomputed on-the-fly (showing different colors). This translates the "important" topics in regard to the selected topic.

Double-clicking on a concept opens a new search stacking the concept query to q_o the original one: $q_o AND(*, x, y)$, $x \neq *$ if the concept is a person, $y \neq *$ if it is a keyword. Clicking on a concept will enact the zoom interaction and subset the list of search results accordingly. Multiple concepts may then be selected. By extension, clicking on any concept in the interface acts similarly. Finally, concepts, or group of concepts may be manually filtered (from a pop-up list), then recompute the visual cloud accordingly.

9 Results

Our original publication [58] has evaluated the visual clouds, since generating this representation was the focus of this paper. In this paper, we present our results from system perspective, through a task validation, a series of usage scenarios and a user study.

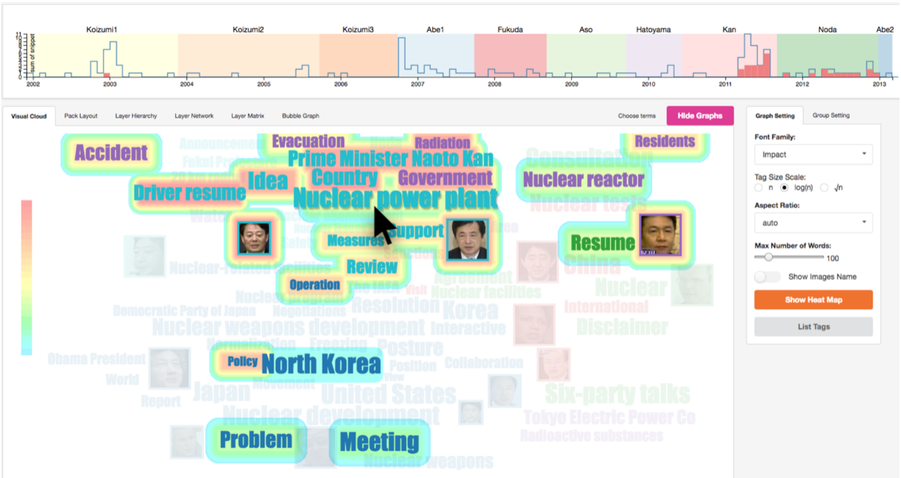


Fig. 18 Hovering a concept highlights its neighborhood in LIN (*i.e.* the leapfrogged network with overlapping layers in M). Heat values are then updated, and the timeline highlighting is linked.

9.1 Task Validation

We may validate our design choices and implementation *w.r.t.* the tasks we have defined in Section 3. We will follow Brehmer *et al.*'s recommendation [11] and their *why?/how?/what?* model to explain how each tasks is performed in our system (*why?* corresponding to the task itself).

*9.1.1 **T1**: Look-up*

User is looking up a specific document or group of documents, knowing some of their properties (*e.g.* people, title, date, *etc.*).
How: (`Introduce`). Show results such that the specific movie is found. **Visual elements:** Search bar. Result list. Result access.
What in: The database, indexed with all documents properties (including full-text indexing).
What out: Details of result.
Process: If a user search a specific movie, or series of movie, she may search the title through the search bar, and the target should appear directly in the list. The complex query enables for the search to be more precise by adding criteria. Changing the ranking criteria in the search list may help further reach the search target. Clicking the result gives access to the actually targeted document.

*9.1.2 **T2**: Locate*

User is looking for the context (*When? Who? What?*) of a specific document (*s.a.* cast or topic of a movie).
How: (`Introduce, Select.`) Access to the details of a specific search result. **Visual**

elements: Search bar. Result list. Result details.
What in: The database, indexed with all documents properties (including full-text indexing).
What out: The details of a results.
Process: Looking for the context of a movie, by knowing how to reach the movie, the user needs to first look-up the movie by typing the right query using the search bar. Once the result is accessed, details of date (*When?*), of people appearing/acting or directing (*Who?*), of keywords/genre (*What?*), as well as summary of the results or other details may be accessed directly from the result presentation in the list.

*9.1.3 **T3**: Browse*

User is looking for some document or groups of documents through information about its context only (*e.g.* crime movies from Woody Allen).
How: (`Introduce`, `Filter`). Show results filtered by some parameters. **Visual elements:** Search bar (Boolean query). Result list. Result access.
What in: The database, indexed by all properties of the movie (including full-text indexing).
What out: A list of a results, details of results.
Process: User uses the Boolean query to fill in contextual information (time-frame, people, keywords) looked for, and may scout from the list the exact searched result(s) to access it(them).

*9.1.4 **T4.1**: Explore: Overview*

User wants to have an idea of how documents distribute over time, what are the other topics important to documents, or who are the main people involve in some documents.
How: (`Introduce`, `Derive`, `Encode`). Show the overall distribution of results filtered by some parameters, across different aspects of the data. **Visual elements:** Search bar (Boolean query). Timeline. Visual cloud.
What in: The database, indexed by all properties of the movie (including full-text indexing).
What out: A map of semantics, a time line.
Process: User uses the Boolean query to fill in partial contextual information (time-frame, people, keywords) looked for. User can now look at distribution over time to find time-frames of interest, or distribution over semantic concepts to find concepts of interest. User may also remove layers through list selection.

*9.1.5 **T4.2**: Explore: Zoom/Filter*

User has found a subset of interest, from the overview and now wants to investigate the perimeter of a selection.
How: (`Select`, `Derive`, `Encode`). Subset a query subgraph depending on overview selection. Derive new overview from selection. **Visual elements:** Timeline. Visual cloud. Hovering. Brushing. Linked highlighting.

What in: A search with overview.
What out: Highlights in Visual Cloud, highlights in time line.
Process: User selects/brushes the timeline, updating the semantic overview representing the potential subset of results. User hovers the visual cloud, updating the timeline distribution and updating the visual cloud from the potential subset of results.

*9.1.6 **T4.3**: Explore: Details-on-demand*

User has confirmed a subset of interest from the overview and now wants to see results/other aspects from a selection.
How: (`Select`, `Filter`, `Navigate`). Subset search results based on selected elements. Derive new overview from selection. **Visual elements:** Result list. New query.
What in: A search with overview.
What out: Subset of search results. New query.
Process: User brushes the timeline, click on a (group of) semantic concepts. It subsets the search results. User double-clicks on a selection of timeline/concepts. It launches a new search with the added selection as a complex criteria. From the search list, detailed search results may be accessed.

9.1.7 Chaining tasks

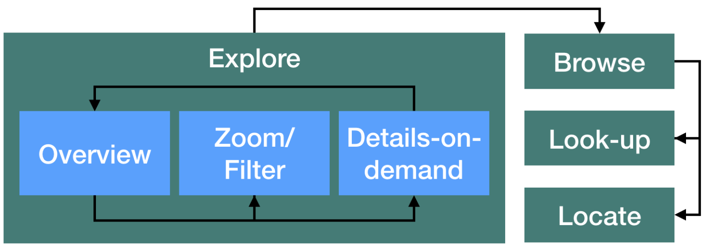


Fig. 19 The different search tasks may be chained one to another.

As we may see from Figure 19, the tasks may be chained. Exploration is an iterative process. The original task may be browsing, but when the location is not absolutely known, user might need a few iterations of overview+filters before finding the right location to look-up for the desired search results. Alternatively, locating tasks may ask for leapfrogged questions that requires exploration.

Such example scenario can take the form: “Who made *this* movie?” (*Locate*) “Now what are the other movies made by the same director?” (*Explore*) and then “What are his movies about, and who else appear in his movies?” (*Explore*), “What was his most sold movie?” (*Browse*), “Let us watch this movie” (*Look-up*).

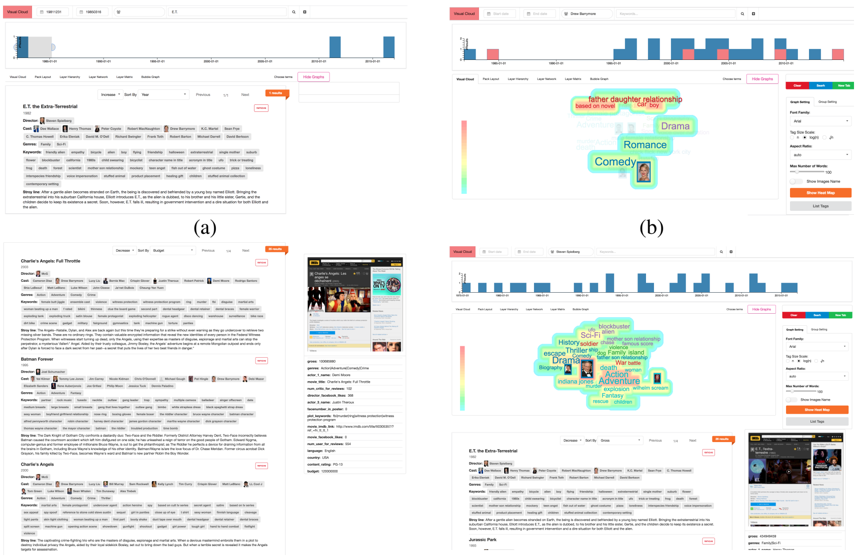
9.2 Usage scenarios

We now present to usage scenario, one for each of our application data, the IMDB5000 dataset and the NHK News archive.

9.2.1 Searching E.T. in IMDB

As suggested in the previous section on chaining tasks (Sec. 9.1.7), we will illustrate how our system can assist in chaining the search tasks.

Let us find the director who made the E.T. movie: we search for *E.T.* in the keywords. Three movies are proposed to us: *E.T. the Extra-Terrestrial*, *Super 8*, and *Fifty Shades of Black*. We refer to the 80's movie and brushing time to this period leaves us only one result: *E.T. the Extra-Terrestrial*. The detailed information shows us that *Steven Spielberg* was the director. To our surprise, we had forgotten that *Drew Barrymore* also played in that movie (Fig. 20(a)).



(a) (b) (c) (d)

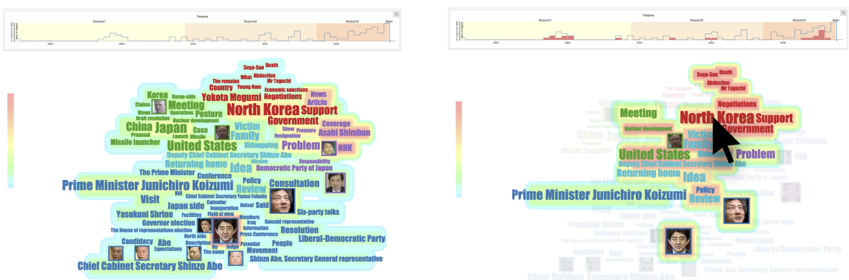
Fig. 20 (a) Finding *E.T.* by brushing the movie in the 80's, confirming that *Steven Spielberg* is the director and noticing that *Drew Barrymore* appeared in this movie. (b) Drew Barrymore's movies about the *father daughter relationship* are often based on *novels*. (c) We sort movies by budget to see the most expensive movie of Drew Barrymore, and from details we can notice it was not bankable as it shows $20M gross less than its budget (*Charlie's Angels: Full Throttle*). (d) Leapfrogging back to Spielberg's movies, his most bankable movie was *E.T.* with a $434M gross. *Harrisson Ford* is also his most recurrent actor thanks to the *Indiana Jones* movie series.

We leapfrog from double clicking *Drew Barrymore* and removing the *E.T.* keyword to investigate her carrier. The timeline shows that she has been very active since the mid-90's. When turning on the visual cloud, we can notice that four clusters form mainly on *Action/Adventure* movies, on *Romance/Comedy*, on *Drama*, and on *father daughter relationship*. Curious about this last aspect, we turn on the heat-map and hover the keyword to this theme has been approached across *Romance/Comedy/Drama* genres, based on *novels* (Fig. 20(b)), mostly since the 2000's (especially *Everybody is Fine*). Clicking on the keyword shows us the list of movies she's acted in about *father daughter relationship*. We may also sort the movies such as discovering her best rated movie from IMDB is *E.T.*, just before *Scream*, using the budget that the two *Charlie's Angels* movies and *Batman Forever* where the most expensive movies she played in (and clicking on details of top two movies, we may see that the second *Charlie's Angels* cost \$20M more than *Batman*, actually bringing back only \$100M, see Fig. 20(c)), and that *E.T.* brought \$434M for a cost of a \$100M when *Batman* only brought \$184M.

We may double-click on *Spielberg* from the *E.T.* result details to leapfrog back to his movies. Sorting the movies by gross confirms that *E.T.* was his most bankable movie ever (Fig. 20(d)). Incidentally, the visual cloud shows that *Harrison Ford* is a redundant actor of Spielberg because of the *Indiana Jones* movies. We finally may go to the IMDB webpage by clicking on the result image of *E.T.*'s result details.

9.2.2 Abe and North Korea on NHK

In a previous study, the authors investigated appearances of Japanese Prime Ministers on NHK [59]. One interesting conclusion was the growth in screen appearances of Abe during Prime Minister Koizumi's ruling (2001/04/26 - 2006/09/26), before becoming himself Prime Minister in the following elections (2006/09/27). Our system allows to refine this study by placing a complex query to search all news segments during Koizumi mentioning Abe by name or face (Fig. 21.a)[4].



(a) (b)

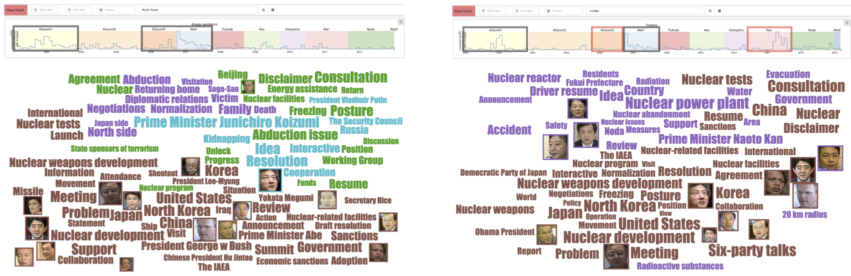
Fig. 21 (a) Interface overview, search query over *Abe* during Koizumi's ruling terms (2001/04/26 - 2006/09/26) very active during the end of *Koizumi3*, as we can see mention of *North Korea* stands out. (b) Hovering over *North Korea* shows that it strongly relates to *abduction*.

The timeline (which is augmented with Prime Minister's rulings on the background) confirms the growing mention of Abe. This is no surprise knowing that Abe was chief cabinet secretary during Koizumi's third term. The visual cloud proposes 5

[4] A demonstration video from our original paper [58] is available at `https://youtu.be/VfGwa6T94t8`.

groups: about elections, about Yasukuni Shrine, about the newspaper Asahi Shimbun, about Japan/Korea/China, and about North Korea. But turning on the heat map, the most prominent word becomes *North Korea* by far. Indeed, Abe was chief negotiator on issues related to abductions of Japanese citizens by North Korea, managing to free 5 of them.

Leapfrogging on the keyword *North Korea* (Fig. 21.b) makes a new search of Abe associated with North Korea during Koizumi terms. Browsing the timeline among the three terms highlights different subtopics at each terms. We may mention a meeting during Koizumi's first term, associated with the three faces of *Koizumi*, *Kim Jong Il*, and *Abe*; and mentions of *sanctions*, *missile launch* and *draft resolutions* during Koizumi's third term.

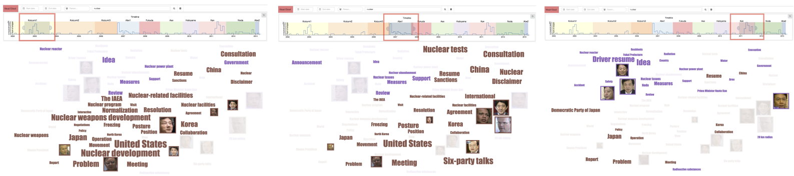


(a) (b)

Fig. 22 Comparison of the queries on *North Korea* (a), and *Nuclear* (b): two periods coincide (in black) Koizumi1 and Abe1 but we can notice two major differences (in red). Koizumi3 period did not associate much *North Korea* and *Nuclear*, and Kan period associate *Nuclear* with *Accident*.

Now a new search on the keyword *North Korea* highlights that it is most active during first Koizumi ruling and especially during first Abe ruling (Fig. 22.a). North Korea related abductions was excessively reported on the media and Abe's administration has put pressure on NHK to "pay attention" [46]. A last search on the keyword *nuclear* gives 3 spikes in the timeline (Fig. 22.b). Two of the spikes coincide with the ones of *North Korea* as we previously described in Figure 22.b. One big difference comes with Koizumi's third term, when no mention of nuclear issue is made. The last in 2011 after the Great East Japan Earthquake about the nuclear power-plant accident.

Each of these spike can be further examined. During the first spike (Fig. 23.a), inspecting the semantic overview, we can see that the faces of *Kim Jong Il* and *Koizumi* are displayed a lot with a focus on North Korea's nuclear development. During the second spike (Fig. 23.b), the faces of foreign leaders in addition to Abe's are often used in association with terms associated to a six-party talk and nuclear tests. During the last period (Fig. 23.c), these are Japanese politicians who are put forward in association with a discussion on power plants and whether or not resuming them.



(a) (b) (c)

Fig. 23 Three perspective on the *nuclear* issue over time. During Koizumi1 (a), focus is on the nuclear development of North Korea. During Abe1 (b), focus is on the 6-party talks. During Kan (c), focus is on resuming nuclear reactors.

9.3 User Study

We wanted to evaluate the usefulness of our overview-rich system in comparison to traditional search interfaces. To do so, we invited 34 people across all age groups (but mostly university students) to evaluate our work. They were asked to use our search engine (with overview), search 10 queries, and compare each with the standard search engine (no overview, just search bar and results, Fig. 24). We interviewed each of them on the usefulness of the interface, took note on their search behavior, and collected informal feed-back. The overall sentiment was globally positive. The results are as follows.

– Most users have used our semantic overview to help search for results when the result they were hoping for was not clearly found within the first two pages of search results.
– When users are not familiar with the area they are searching in (location unknown), they prefer using our interface. For example some users preferred using our interface to search about celebrities.
– All users are interested in the interactive re-ranking of results and would like it implemented in regular search engines.
– They globally found our visual cloud with highlight more readable than their own experience of common tag cloud. However, they only showed more interest to the heat-map as the number of results (and words) was increasing.
– 17 users have expressed that they would like to use our interface in the future, 10 users will use our interface under some circumstances, 7 have expressed no interest whatsoever.

9.4 Implementation and Complexity

Exploratory tasks require a lot of successive iterations during search, that of course rely on interaction. In addition, we desired our system to be running within a regular web browser. This implies for our algorithms to be fast enough so changes may be instantaneously reflected to users (*i.e. on-the-fly*).

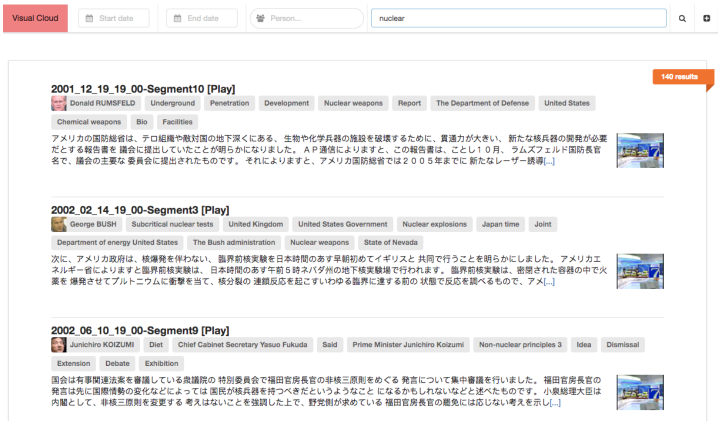


Fig. 24 Standard search interface.

The system[5] is implemented in HTML5 (with canvas for the visual cloud) with the popular javascript libraries semantic-UI and d3 libraries[6]. Database indexing and access is implemented on server side in python.

The construction of the *LIN* and $n_{l,l'}$ are made while constructing the multilayer network of results. Both depend on the number of results $|V|$ and number of concepts $|L|$ and this data abstraction construction runs in $O(\frac{1}{2}(|V|*(|V|-1)+|L|*(|L|-1))$. Note that in practice, the number of layers $|L|$ reaches the 10^2 order of magnitude, and we limit results $|V|$ to the order 10^3. These assumptions may be reasonable, given the necessary readability of the semantic overview and the size of a search result space in our application context.

The computation of entanglement values is bounded by an eigen decomposition of a square matrix of dimension based on the number of layers $|L|$, estimated to be $O(|L|^{2.37})$ at maximum [18]. The computation of the hierarchy is a greedy optimization similar to Louvain [10] estimated in $O(|L|log|L|)$. Note that our implementation of the hierarchy is stable by definition (does not imply any stochastic process) and can be recomputed instantaneously.

No complexity is discussed for the Pack-Layout algorithm [74] but its computation completes within milliseconds for a thousand circles. The number of circles depends on $|L|$, which is at best a few hundreds. The word cloud generation is based on Davies' heavily optimized implementation[7], which is bounded by computation of bounding boxes and collisions (not impacted by our modifications). The heat-map generation is done in one pass through each concept, with a static Canvas implementation faster and more memory efficient than DOM population of SVG elements. Including the Pack-Layout initialization, the word cloud generation, heatmap, and any selection/hovering runs below the second and can be considered instantaneous.

[5] Available at `https://github.com/renoust/visualclouds`
[6] See `https://semantic-ui.com/` and `https://d3js.org/`
[7] See `www.jasondavies.com/wordcloud/about/`

10 Discussion and Conclusion

While our initial paper [58] was focused on the construction of the visual cloud semantic overview, we have introduced in this paper how multilayer networks, as a data abstraction, can support all search tasks across heterogeneous criteria through a visual analytics system. After introducing design requirements for our search system, we have introduced two datasets, a query system and how we model query results as a multilayer network. We have then detailed manipulations of the multilayer network and elements of analysis, so we may highlight salient characteristics of the results, and extract hierarchies and groups. With this toolbox in hands, we then introduced how we implemented our system, first with visual encoding, then with interactions. We finally evaluated our system in response of design to tasks validation, through usage scenarios, with a user study, and discussed complexity.

In this work, we have used multilayer networks as a backbone to support all our tasks, in addition to data construction, manipulation, and visualization, all done without *actually* displaying the multilayer network itself. There are still many aspects of the multilayer networks that we have left unexplored here. For example, we could use an aggregated layer (as in Fig. 6) or use a t-SNE [43] style 2D embedding to display a density map to help drive users. We have not explored either all analytical possibilities that multilayer networks offer (*s.a.* layer compression) and we keep them for our future work.

One limitation of our system is the scalabilty of overview in terms of results. We do not give any overview over millions of results but only on the top few hundreds (since users only consult the first few pages [69] [70]). This fits well our application context, but if we were to deliver millions of results, we should think of the representativeness of our overviews. One approach could lead to clustering of the results and exploring these clusters remain a task we have not tackled yet.

This system only articulates two aspects of the data: time and concepts. We have not considered to include location information yet. It would be extremely relevant to propose a geographical perspective that would be linked highlighted similarly to time and concepts. It would relieve the semantic view from location references. For example, one could imagine a geographical heat map to show geographical context of search result. Such extension would display the generic nature of our approach by extending its range of applications.

To further extend browsing and exploration tasks, we would like to add a nearest neighbor hyperlinking between search results. The nearest neighbors also form a multilayer network that may be overviewed in different aspects. It would enable further leapfrogged explorations, but this time, leapfrogged from the perspective of the multimedia documents.

We have improved traditional tag clouds with images as a supplementary information, bringing new information in form of visual cues. Integrating visual objects in a tag cloud fashion seems a promising way. We have only used faces this time but we plan to use other cues, such as objects, logos, and, most importantly, frequent images. The heat map being an interesting addition for exploration-related tasks, we plan to further dig into its design by providing a customizable metric to the heat values. We could also think of constraining further the Visual Cloud in order to see both topical grouping and time evolution of topic (*e.g.* in a Theme River fashion [26]).

One last important future work concerns *comparison* tasks. We currently refine information through visual cloud hovering, timeline browsing, and leapfrogging. However, beyond side-by-side comparison of two queries tabs, we do not have explored other means of comparison. This need quickly rises as we would like to compare periods of time. From the relationship between interactions and subgraphs, we believe the multilayer abstraction could provide interesting manipulations suited for comparison.

Acknowledgement

We would like to dedicate this work in memory of our co-author Marie-Luce Viaud, who left us too soon during the process of completing this work.

References

1. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visualizing time-oriented dataa systematic view. Computers & Graphics **31**(3), 401–409 (2007)
2. Amar, R., Stasko, J.: A knowledge task-based framework for design and evaluation of information visualizations. In: Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, pp. 143–150. IEEE (2004)
3. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: A general-purpose face recognition library with mobile applications. Tech. rep., Technical report, CMU-CS-16-118, CMU (2016)
4. Archambault, D., Munzner, T., Auber, D.: Grouseflocks: Steerable exploration of graph hierarchy space. IEEE transactions on visualization and computer graphics **14**(4), 900–913 (2008)
5. Azzopardi, L., Kelly, D., Brennan, K.: How query cost affects search behavior. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 23–32. ACM (2013)
6. Baeza-Yates, R., Davis, E.: Web page ranking using link attributes. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 328–329. ACM (2004)
7. Barry, A.M.: Visual intelligence: Perception, image, and manipulation in visual communication. SUNY Press (1997)
8. Barth, L., Kobourov, S.G., Pupyrev, S.: An experimental study of algorithms for semantics-preserving word cloud layout. University of Arizona Report (2013)
9. Becker, R.A., Cleveland, W.S.: Brushing scatterplots. Technometrics **29**(2), 127–142 (1987)
10. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **10**, 8 (2008). DOI 10.1088/1742-5468/2008/10/P10008
11. Brehmer, M., Munzner, T.: A multi-level typology of abstract visualization tasks. IEEE TVCG **19**(12), 2376–2385 (2013)
12. Brin, S., Page, L.: Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks **56**(18), 3825–3833 (2012)
13. Burt, R., Scott, T.: Relation content in multiple networks. Social Science Research **14**, 287–308 (1985)
14. Carpineto, C., Osiński, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Computing Surveys (CSUR) **41**(3), 17 (2009)
15. Chinchor, N.A., Thomas, J.J., Wong, P.C., Christel, M.G., Ribarsky, W.: Multimedia analysis+ visual analytics= multimedia analytics. IEEE computer graphics and applications **30**(5), 52–60 (2010)
16. Clarkson, E., Desai, K., Foley, J.: Resultmaps: Visualization for search interfaces. IEEE TVCG **15**(6) (2009)

17. De Domenico, M., Porter, M.A., Arenas, A.: Muxviz: a tool for multilayer analysis and visualization of networks. Journal of Complex Networks **3**(2), 159–176 (2015)
18. Demmel, J., Dumitriu, I., Holtz, O.: Fast linear algebra is stable. Numerische Mathematik **108**(1), 59–91 (2007)
19. Di Marco, A., Navigli, R.: Clustering and diversifying web search results with graph-based word sense induction. Computational Linguistics **39**(3), 709–754 (2013)
20. Ferragina, P., Gullì, A.: Knowledge Discovery in Databases: PKDD 2004, chap. The Anatomy of SnakeT: A Hierarchical Clustering Engine for Web-Page Snippets, pp. 506–508. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). DOI 10.1007/978-3-540-30116-5_48
21. Fujimura, K., Toda, H., Inoue, T., Hiroshima, N., Kataoka, R., Sugizaki, M.: Blograngera multi-faceted blog search engine. In: Proc. WWW, Weblogging Ecosystem (2006)
22. Gao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Robustness of a network of networks. Physical Review Letters **107**(19), 195,701 (2011)
23. Ghoniem, M., Fekete, J.D., Castagliola, P.: A comparison of the readability of graphs using node-link and matrix-based representations. In: IEEE Symposium on Information Visualization, pp. 17–24. Ieee (2004)
24. Ghoniem, M., Mcgee, F., Melançon, G., Otjacques, B., Pinaud, B.: The state of the art in multilayer network visualization. Computer Graphics Forum (2019). DOI 10.1111/cgf.13610
25. Gomez-Nieto, E., San Roman, F., et al.: Similarity preserving snippet-based visualization of web search results. IEEE TVCG **20**(3), 457–470 (2014)
26. Havre, S., Hetzler, B., Nowell, L.: Themeriver: Visualizing theme changes over time. In: IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings, pp. 115–123. IEEE (2000)
27. Hervé, N., Viaud, M.L., Thièvre, J., Saulnier, A., Champ, J., Letessier, P., Buisson, O., Joly, A.: Otmedia: the french transmedia news observatory. In: Proc. ACM Multimedia, pp. 441–442. ACM (2013)
28. Ide, I., Nack, F.: Explain this to me! ITE Trans. on Media Technology and Applications **1**(2), 101–117 (2013)
29. Ide *et al.*, I.: Topic threading for structuring a large-scale news video archive. Image and Video Retrieval **1**(1), 123–131 (2004)
30. Itoh, M., Toyoda, M., Zhu, C.Z., Satoh, S., Kitsuregawa, M.: Image flows visualization for inter-media comparison. In: IEEE Pacific Visualization Symposium, 2014, pp. 129–136. IEEE (2014)
31. Käki, M.: Findex: Search result categories help users when document ranking fails. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 131–140 (2005)
32. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. Journal of complex networks **2**(3), 203–271 (2014)
33. Kochtchi, A., Landesberger, T.v., Biemann, C.: Networks of names: Visual exploration and semi-automatic tagging of social networks from newspaper articles. In: Computer Graphics Forum, vol. 33-3, pp. 211–220. Wiley Online Library (2014)
34. Koshman, S.: Visualization-based information retrieval on the web. Library & Information Science Research **28**(2), 192–207 (2006)
35. Koshman, S., Spink, A., Jansen, B.J.: Web searching on the vivisimo search engine. Journal of the American Society for Information Science and Technology **57**(14), 1875–1887 (2006)
36. Krovetz, R., Croft, B.: Lexical ambiguity and information retrieval. ACM Trans on Information Systems (TOIS) **10**(2), 115–141 (1992)
37. Krstajic, M., Najm-Araghi, M., Mansmann, F., Keim, D.A.: Incremental visual text analytics of news story development. In: Visualization and Data Analysis, p. 829407 (2012)
38. Le, D.D., Phan, S., Nguyen, V.T., Renoust, B., Nguyen, T.A., Hoang, V.N., Ngo, T.D., Tran, M.T., Watanabe, Y., Klinkigt, M., et al.: Nii-hitachi-uit at trecvid 2016. In: TRECVID Workshop. NIST, Gaithersburg, MD, USA. (2016)
39. Le, D.D., Satoh, S.: Indexing faces in broadcast news video archives. In: 2011 IEEE 11th ICDM Workshops, pp. 519–526 (2011)
40. Le, T.N., Luqman, M.M., Burie, J.C., Ogier, J.M.: A comic retrieval system based on multilayer graph representation and graph mining. In: International Workshop on Graph-Based Representations in Pattern Recognition, pp. 355–364. Springer (2015)
41. Li, H., Jou, B., Ellis, J.G., Morozoff, D., Chang, S.F.: News Rover: Exploring topical structures and serendipity in heterogeneous multimedia news. In: Proc. Multimedia, pp. 449–450. ACM (2013)
42. Luo, H., Fan, J., Yang, J., Ribarsky, W., Satoh, S.I.: Exploring large-scale video news via interactive visualization. In: IEEE VAST, pp. 75–82. IEEE (2006)

43. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
44. Marchionini, G.: Exploratory search: from finding to understanding. Communications of the ACM **49**(4), 41–46 (2006)
45. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based word clustering using a web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 542–550. Association for Computational Linguistics (2006)
46. McCormack, G.: Japan and north korea: The long and twisted path toward normalcy. Tech. rep., Working Paper. US-Korea Inst. at SAIS (2008)
47. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
48. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. Journal of Visual Languages & Computing **6**(2), 183–210 (1995)
49. Moltmann, F.: Natural language ontology. Oxford Research Encyclopedia of Linguistics (2017). DOI 10.1093/acrefore/9780199384655.013.330
50. Munzner, T.: A nested model for visualization design and validation. IEEE transactions on visualization and computer graphics **15**(6), 921–928 (2009)
51. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys **41**(2), 10 (2009)
52. Ngo *et al.*, T.D.: Face retrieval in large-scale news video datasets. IEICE Trans. on Information and Systems **96**(8), 1811–1825 (2013)
53. Noack, A.: Modularity clustering is force-directed layout. Physical Review E **79**(2), 026,102 (2009)
54. Nocaj, A., Brandes, U.: Organizing search results with a reference map. IEEE TVCG **18**(12), 2546–2555 (2012)
55. Osiński, S., Weiss, D.: Carrot2: Design of a flexible and efficient web information retrieval framework. In: International Atlantic Web Intelligence Conference, pp. 439–444. Springer (2005)
56. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
57. Peat, H.J., Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems. Journal of the american society for information science **42**(5), 378–383 (1991)
58. Ren, H., Renoust, B., Viaud, M.L., Melançon, G., Satoh, S.: Generating visual clouds from multiplex networks for tv news archive query visualization. In: 2018 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE (2018)
59. Renoust, B., Kobayashi, T., Ngo, T.D., Le, D.D., Satoh, S.: When face-tracking meets social networks: a story of politics in news videos. Applied Network Science **1**(1), 4 (2016)
60. Renoust, B., Melancon, G., Munzner, T.: Detangler: Visual analytics for multiplex networks. In: Computer Graphics Forum, vol. 34-3, pp. 321–330. Wiley Online Library (2015)
61. Renoust, B., Melançon, G., Viaud, M.L.: Entanglement in multiplex networks: understanding group cohesion in homophily networks. In: Social Network Analysis, pp. 89–117. Springer (2014)
62. Scaiella, U., Ferragina, P., Marino, A., Ciaramita, M.: Topical clustering of search results. In: Proc. WSDM, pp. 223–232. ACM (2012)
63. Selberg, E., Etzioni, O.: Multi-service search and comparison using the metacrawler. In: In Proceedings of the 4th International World Wide Web Conference, pp. 195–208 (1995)
64. Shi, J., Tomasi, C.: Good features to track. In: Proceedings CVPR'94., pp. 593–600. IEEE (1994)
65. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Visual Languages, pp. 336–343. IEEE (1996)
66. Sinclair, J., Cardew-Hall, M.: The folksonomy tag cloud: when is it useful? Journal of Information Science **34**(1), 15–29 (2008)
67. Singer, J.B.: Five ws and an h: Digital challenges in newspaper newsrooms and boardrooms. The International Journal on Media Management **10**(3), 122–129 (2008)
68. Spink, A., Jansen, B.J., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. Computer **35**(3), 107–109 (2002)
69. Spink, A., Wolfram, D., Jansen, M.B., Saracevic, T.: Searching the web: The public and their queries. Journal of the American society for information science and technology **52**(3), 226–234 (2001)
70. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 415–422. ACM (2004)
71. Viegas, F.B., Wattenberg, M., Feinberg, J.: Participatory visualization with wordle. IEEE TVCG **15**(6), 1137–1144 (2009)

72. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)
73. Wang, M., Li, H., Tao, D., Lu, K., Wu, X.: Multimodal graph-based reranking for web image search. IEEE Transactions on Image Processing **21**(11), 4649–4661 (2012)
74. Wang, W., Wang, H., Dai, G., Wang, H.: Visualization of large hierarchical data by circle packing. In: Proc. SIGCHI, CHI '06, pp. 517–520. ACM, New York, NY, USA (2006). DOI 10.1145/1124772.1124851
75. Wasserman, S., Faust, K.: Social Network Analysis, Methods and Applications. Structural Analysis in the Social Sciences. Cambridge University Press (1994). DOI 10.1017/CBO9780511815478
76. Wilson, M.L., Kules, B., Shneiderman, B., et al.: From keyword search to exploration: Designing future search interfaces for the web. Foundations and Trends® in Web Science **2**(1), 1–97 (2010)
77. Wu, Y., Provan, T., Wei, F., Liu, S., Ma, K.L.: Semantic-preserving word clouds by seam carving. Computer Graphics Forum **30**(3), 741–750 (2011)
78. Xu, J., Tao, Y., Lin, H.: Semantic word cloud generation based on word embeddings. In: PacifVis,, pp. 239–243 (2016)
79. Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., Ma, W.Y.: Improving web search results using affinity graph. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 504–511. ACM (2005)
80. Zhang, D., Dong, Y.: Advanced Web Technologies and Applications: 6th Asia-Pacific Web Conference, APWeb 2004, chap. Semantic, Hierarchical, Online Clustering of Web Search Results, pp. 69–78. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). DOI 10.1007/978-3-540-24655-8_8
81. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proc. CIKM, CIKM '02, pp. 515–524. ACM, New York, NY, USA (2002). DOI 10.1145/584792.584877