Temporal Visualization of a Multidimensional Network of News Clips

Filipe Gomes, José Devezas, and Álvaro Figueira

CRACS/INESC TEC, Faculdade de Ciências, Universidade do Porto
Rua do Campo Alegre, 1021/1055, 4169-007 Porto, Portugal
{filipe.gomes,jld,arf}@dcc.fc.up.pt

Abstract. The exploration of large networks carries inherent challenges in the visualization of a great amount of data. We built an interactive visualization system for the purpose of exploring a large multidimensional network of news clips over time. These are clips gathered by users from web news sources and references to people or places are extracted from. In this paper, we present the system's capabilities and user interface and discuss its advantages in terms of the browsing and extraction of knowledge from the data. These capabilities include a textual search and associated event detection, and temporal navigation allowing the user to seek a certain date and timespan.

Keywords: visualization, networks, search, event detection

1 Introduction

The exploration of large networks, using force-directed graphs [9], carries inherent challenges in the visualization of a great amount of data. One problem arises, firstly, from the large number of nodes present in the graph and, secondly, from the potentially exponential growing number of links, that clutter the screen. Associated to this problem is the smooth navigation of the graph, which is addressed with an intuitive user interface and a reduced visual disturbance in the graph's deployment.

In this paper, we describe a visualization and browsing system for a network of news clips with three dimensions for the edges — *who*, *where* and *when* — and temporal annotations for each node. This network is based on a project with the goal of connecting online news and the Social Web. The network is based on the relations induced by the coreference of entities between a compilation of news clips taken from online news sources.

2 Reference Work

Previous studies have reported that visualization has increasingly been used in data mining. While data generation has been growing exponentially, the exploration of large data sets has become a prominent but challenging problem, as

visual data mining is an interesting instrument handle this data deluge [11]. A visual data mining system must be syntactically simple to be useful, that is, it must have intuitive user interaction, a clear output and, thus, should be considered a powerful tool [14].

In recent years, there has been a considerable amount of literature published on the topic of large network visualization. Previous works have been undertaken, presenting visual analytics tools [13], having been identified as breakthroughs in data exploration in the context of large data sets [10]. The development of these tools analysed significant aspects of visualization tools, such as the usability or scalability [1]. Moreover, the exploration through time of a network carries its own challenges: new data should be presented in context of older data and the visualization must be dynamic for the user to realize the changes [12].

3 Data Set

Our system resorts to a collection of news clips that, for the purpose of this paper, were fetched automatically from select news outlets' websites. We developed a crawler that is scheduled to run twice daily, to gather the top news from successive days. The goal is to artificially generate, from the crawled news articles, a news clips collection similar to a user-generated one. To do that, we segmented each news article into sentences and automatically generated between two and five clips, with two to seven sentences each, by randomly selecting an initial starting sentence as well as the referred parameters. The gathered data set contains 8,981 clips, fetched between the 8^{th} and 12^{th} November 2012 for ten different news sources: Euronews, United Press International, BBC News, Daily Mail, Guardian, Reuters, The Telegraph, USA Today, New York Times, and Washington Post. News clips size varies between 1 and 376 words, with an average of 102 words. Using the methodology described in Devezas and Figueira [6], we extracted the named entities from this data set, including people, places and dates. Table 1 shows the top referenced people and places in our collection. Dates were not included in the table in order to make topic distinction simpler.

Table 1: Top 20 people and places identified by our system within the data set.

Rank	Entity	Frequency	Rank	Entity	Frequency
1	China	252	11	David Cameron	83
2	England	176	12	Scotland	82
3	Syria	148	13	Jersey	78
4	Wales	137	14	Europe	71
5	United States	123	15	Africa	71
6	Afghanistan	111	16	Israel	67
7	Iraq	88	17	France	66
8	Barack Obama	85	18	Mitt Romney	61
9	India	84	19	Turkey	53
10	Russia	84	20	Mexico	50

3.1 Multidimensional Network of News Clips

We built a news clips network based on the coreference of the identified entities across text fragments. For each type of entities (people, places and dates), we established a distinct dimension, using three different edge types: *who*, *where* and *when*. This resulted in a multidimensional network with 8,981 nodes representing news clips and 323,177 edges representing a connector entity referenced in the pair of news clips it linked.

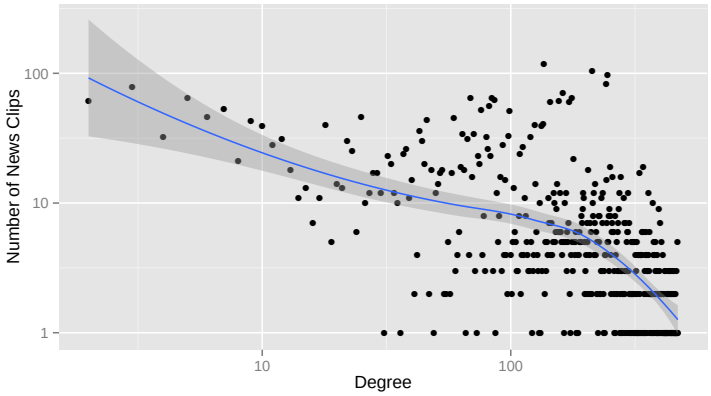


Fig. 1: Degree distribution of the news clips network.

A brief analysis of the node degree distribution, depicted in Fig. 1, shows a heavy-tail behaviour, which indicates the presence of a scale-free network. As we can see from Table 2, the multidimensional network of news clips also presents the property of small-world, having a higher clustering coefficient and a lower average shortest path length than its random counterpart, as generated by the Erdős–Rényi model [8] using the same number of nodes and edges.

Table 2: Comparison of the original network with its random counterpart.

Network	Clustering Coefficient	Average Path Length
Random	0.008006131	2.549247
Original	0.590759899	2.493203

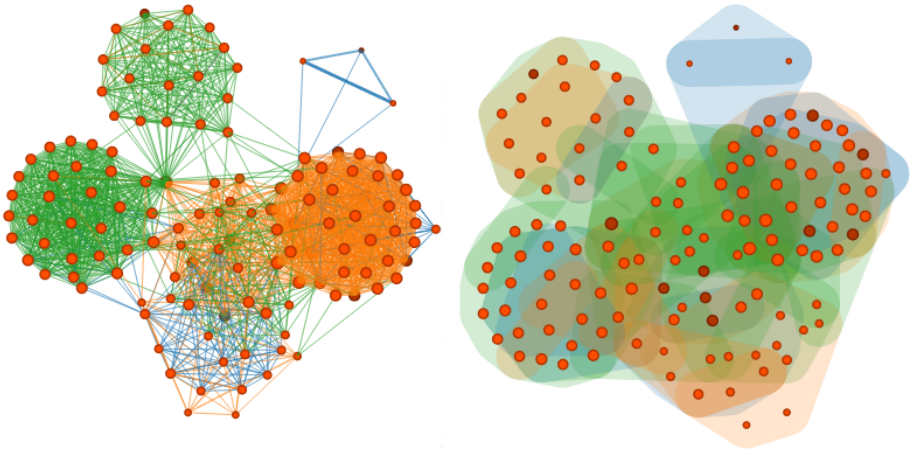
4 Visualization System

In this section, we present the visualization system, firstly introduced in [7], on which our system is built upon. We extended the system by allowing the topical

and temporal search of the news clips' database. Thus, the system indexes news clips, as described in Section 4.2. Moreover, we developed an event detection system that we present in Section 4.3. Finally, we present the system as a whole, with the navigation features that depend on the aforementioned components.

4.1 Multidimensional Network Visualization

We built our multidimensional network visualization system on top of the system presented in [7]. Ergo, it shares the same foundation, being developed using the D3.js JavaScript library [4]. The graph data is transferred using the XML-based GraphML format [5]. Data is transferred asynchronously from the requests and consists only in a relevant portion of the graph, that is shown to the user.



(a) Traditional lines for edges. (b) Convex hulls replacing edges.

Fig. 2: Comparing the visualization of the network using lines for edges or convex hulls to group nodes that reference a common entity.

As shown in Fig. 2, the graph displayed by the visualization system is a force-directed graph. While on Fig. 2a we can see a typical illustration of a network depicted as a graph, on Fig. 2b the nodes are laid out according to their links but those links are not shown to the user.

We chose to hide the links as it allows for less graphical complexity, and to display convex hulls. This is possible for this kind of network, since every node referencing a given entity will be interconnected, thus forming a complete subgraph, that can be contained within a visual group depicted by a convex hull. The hulls' color is determined according to the entity's dimension. Due to the hulls transparency, the number of different relations and nodes involved are easily perceived. Albeit not flawless with great number of relations, the alternative is even less clear with the sheer concentration of edges.

Table 3: Comparison of a graph rendering, with visible edges or hulls.

Type	Num. Nodes	Num. Edges	Iterations	Avg. Frame Time	Total Time
Edges	301	14642	298	263.1 ms	78.4 s
Hulls				11.0 ms	3.3 s

The contrast between the two cases is flagrant, particularly in large graphs, as the experimental data shown in Table 3 attests. The average time it takes to compute each frame is nearly 24 times higher, with the edges visible, taking a longer time for the graph to settle down. Moreover, when such large times, like the one obtained with a rather large graph, are experienced, the user will find the application slow and unresponsive.

4.2 Index and Search

We also address the problem of visual clutter, by providing the users with the capability of analysing a selected group of news clips that are relevant to them. To do so, we index the news clips and prepare a search engine that enables generic queries as well as queries within a specified timespan or for a given user.

We take advantage of the Apache Solr open source enterprise search platform [3], built upon the Apache Lucene project [2], to filter, index and search the Breadcrumbs news clips collection. For each news clip, we index the following fields: clip ID, text fragment, title, comment, domain name, owner username, creation date and tags. We also include a field *all*, that is used by default in a query, which results of the concatenation of the text fragment, title, comment and tags of a news clip. This way, the default behavior is to include all of the relevant information in a generic search, broadening the result space (e.g. a news clip with the tag "obama", that doesn't reference Barack Obama in the text, can be returned). In the Solr index and query schema, we define a *text_general* field type, used by all text fields, except the user and the domain names, which tokenizes the text with Lucene's *StandardTokenizer* and filters it in order to remove stop words from a custom bilingual dictionary (English and Portuguese) and to convert the text to lower case. Separate but similar filter groups are defined for the index and query phases, the difference being the addition of a synonym filter during the query phase.

An instance of Apache Solr is deployed to our application server, running alongside the Breadcrumbs application. Communication is established using a custom wrapper service implemented in the Breadcrumbs system, responsible for directly querying Solr, which is only available in the server-side, as a web service.

4.3 Event Detection

At an initial development stage of the visualization system, we identified a usability issue regarding the temporal navigation of the news clips network. Whenever

there were no news clips for a selected time interval, a user would be presented with an empty canvas, having to guess a valid interval where some of the collected news clips would be displayed. In order to solve this problem, we developed an event detection subsystem, based on the detection of relevant peaks in a time series depicting the number of results of a given search for each day.

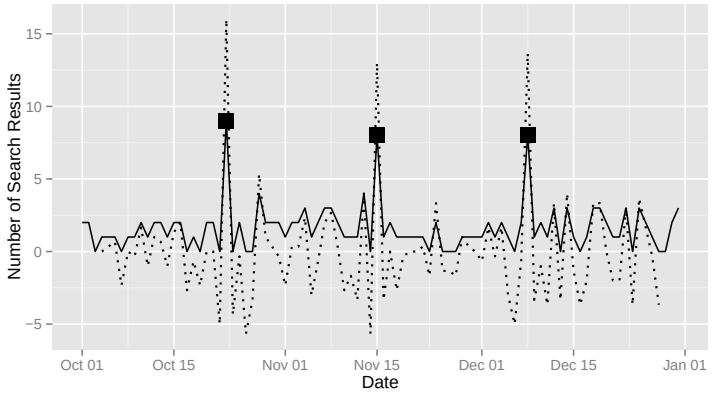


Fig. 3: Example time series depicting the number of news clips, for each day, retrieved for a hypothetical query. The solid line represents the time series itself, while the dotted line represents the event strengths, calculated by our algorithm, and the square points represent the identified events.

Our algorithm is based on the idea that a peak should only represent an event for the given query whenever its value is well above average when compared the number of news clips for each of the remaining days. Let's assume an hypothetical scenario where a user would search for a given query (e.g. "obama romney election") and a list of dated results would be returned. Fig. 3 shows a time series, illustrating the distribution over time of search results, within the range of returned dates, filling days without results with the value zero. For each known value, our algorithm calculates the event strength, a score that helps determine the probability of a peak corresponding to a relevant event. The event strength for a given point t_i depends on the value of the point y_i as well as the left and right values, y_ℓ and y_r respectively, where $\ell \in [i-n, i-1]$ and $r \in [i+1, i+n]$, for a given window of size n. In this specific example, we used $n = 3$. Event strength E_{str} is calculated as described in Equation 1.

$$E_{str} = 2 \times y_i - \left(\sum_{\ell=i-n}^{i-1} y_\ell + \sum_{r=i+1}^{i+n} y_r \right) \tag{1}$$

The event strength values for each point are also depicted in Fig. 3 using a dotted line. As we can see, irrelevant points tend to have negative or very

low values, while relevant points have a much higher event strength. In order to retrieve the event points, we now need to define a threshold above which we accept a peak as an event. Notice that some of the low value peaks, also have a positive value for the event strength, however they aren't much different from their neighbourhood and shouldn't therefore be considered a relevant event. On the other hand, there are three peaks (marked with a square) that depict an out of the ordinary number of results, for the given query, on a single day. Each of these peaks has an event strength that is much higher than the peak's value. This characteristic is unique to relevant events. Thus, an event for the requested search is identified in a point t_i whenever, $E_{str}^{(i)} - y_i > \sigma_Y$, where $E_{str}^{(i)}$ is the event strength for point t_i and σ_Y is the standard deviation for the time series values.

4.4 Graph Navigation

Provided a topic and a timespan, the system fetches a given number of clips. These are selected, searching the keywords in the title, tags or body of each news clip. Furthermore, the user can specify in which field to search for a given keyword. In this way, the system is able to filter the clips quite narrowly, in accordance with the user's indications. The resultant clips are presented as the initial nodes to the user, and make up the kernel for the forthcoming graph. The system builds the graph with joining the neighbour clips: clips that have a relationship with any of the searched ones, that is, based on the people, places and dates they may both reference.

The graph itself is built iteratively, expanding with each consecutive user action. Thus, when the user enters a new topic or changes the timespan, the browser requests a list of clips for the given topic, over the specified timespan, as we described before. These are presented on the screen as unconnected nodes. The browser than requests the server a graph that contains these last clips and their neighbours.

In order to maintain an trimmed graph, some nodes are gradually withdrawn from the graph, being selected based on two factors: the clip's date and the node's order of entry in the graph. The latter is a simple threshold to prevent the overcrowding of the screen. Beyond maintaining a visually engaging graph, this factor contributes to the reducing of the computational complexity of graph visualization, as less elements need to be positioned on the screen. On the other hand, the nodes whose news clip's date is no longer over the selected timespan are the ones more commonly removed, ensuring a certain consistency to the displayed graph, as only the specified timespan is shown to the user.

When a new search is performed, the system fetches the list of events for the given topic. As depicted in Fig. 4, the user inserts the search terms in (A). The events are displayed in the timeline, as markers (E) under the slider bar. The marker's tone is proportional to the intensity of the event.

In order to facilitate the navigation in a wide time range, the user can shift the slider's lower and upper bounds (D). This allows for a more or less extended view of any time range. Anticipating a substantial use of the slider, while the

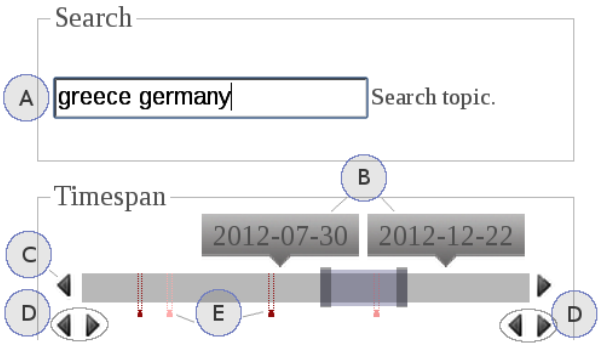


Fig. 4: System's parameter configuration: giving a search input and a timespan.

user is searching for a given event or exploring the evolution of different clip clusterings, this element received special attention. Accordingly, the user has various interactions methods available, the slider can be moved by clicking the arrows aside it (C), dragging its limits or date labels (B), or using the mouse wheel.

The graph is interactive, so the user can move nodes around to improve the browsing experience. Moreover, the user can pinpoint and, with a mouse click, expand a node, that is, display its neighbours within the selected timespan. This feature allows for a loosely and rather precise exploration of the graph, potentially enabling the discovery of supplementary clips.

5 Discussion

To demonstrate the applications of our multidimensional network visualization, we present some use-cases. The combination of our system's different features allows for easier information retrieval and browsing, providing insight into the knowledge underlying in the data.

The starting point, when using the system, is the search query and the timespan. The latter is quite straightforward, the user can narrow the browsing in a timespan more or less confined according to the knowledge of a relevant exact date or the larger period when a series of events might have happened. Regardless, the time span can be shorten in the case of a rather wide period, or the timespan can be readily scrolled until interesting clips emerge.

Browsing the network through time allows the user to analyse sequence of events in the news. Taking, for example, news clips related to Israel. In the wake of the US elections, clips discuss the relation between the two countries and the election's outcome on Israel's politics. In the following days, the focus shifts back to the ongoing conflict in Syria and references to countries in the region emerge. Finally, news clips arise relating to the Gaza conflict, foretelling its escalation in the following days. The temporal navigation allows the user to

sift through these different but related events and to perform a detailed analysis in that period context.

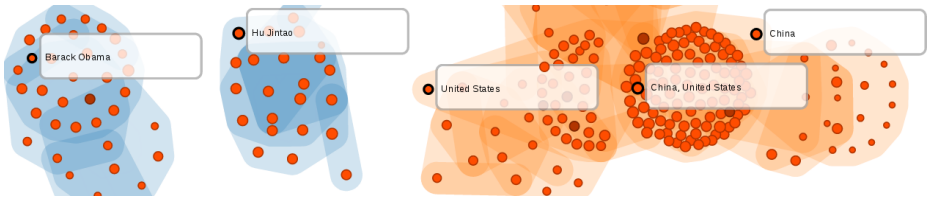


Fig. 5: Comparison between different co-reference dimensions.

Turning to the visualization itself, the user is able to perform a broad analysis of the information presented. Taking an example concerning the United States and China, both undergoing a power transition at the time of the query. The resulting graph for the two countries or leaders will show them connected by the country entities, as shown in the right graph in Fig. 5. This is expected due to geopolitical factors and economical ties between the two. Nonetheless, if we filter out spacial references leaving only the entities from the class *People*, we note a much thinner, or no connection at all, between the two communities.

6 Conclusions and Future Work

In this paper, we present our temporal visualization system of a multidimensional network, with search and temporal navigation features, including an event detection system. We propose a distinct force-directed graph visualization where the edges are hidden, instead displaying convex hulls covering nodes co-referencing the same entity. This reduces the graph's rendering computational complexity and visual clutter. Our system enables the user to browse a large network of news clips spanning a wide timespan. Beyond browsing, the listed key features play an important role in the extraction of knowledge, providing an intuitive and effective user interface.

Future work in this area should focus on the scalability. Further developments would be needed in the algorithms in order to handle relatively large graphs, that may arise even for small a timespan.

Acknowledgements

This work is financed by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-C/EEI/LA0014/2011. We would like to thank Nuno Cravino for the helpful discussions, specially regarding the event detection algorithm.

References

1. Abello, J., van Ham, F., Krishnan, N.: Ask-graphview: A large scale graph visualization system. Visualization and Computer Graphics, IEEE Transactions on 12(5), 669 –676 (sept-oct 2006)
2. Apache Software Foundation: Apache Lucene. http://lucene.apache.org/ (2001), retrieved November 9, 2012
3. Apache Software Foundation: Apache Solr. http://lucene.apache.org/solr (2006), retrieved November 9, 2012
4. Bostock, M., Ogievetsky, V., Heer, J.: D-3: Data-Driven Documents. IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS 17(12), 2301–2309 (DEC 2011), IEEE Visualization Conference (Vis)/IEEE Information Visualization Conference (InfoVis), Providence, RI, OCT 23-28, 2011
5. Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., Marshall, M.S.: Graphml progress report - structural layer proposal (2002)
6. Devezas, J., Figueira, A.: Finding Language-Independent Contextual Supernodes on Coreference Networks. IAENG International Journal of Computer Science 39(2), 200–207 (2012), http://www.iaeng.org/IJCS/issues_v39/issue_2/IJCS_39_2_07.pdf
7. Devezas, J., Figueira, A.: Interactive Visualization of a News Clips Network: A Journalistic Research and Knowledge Discovery Tool. In: Proceedings of the 4th International Conference on Knowledge Discovery and Information Retrieval (KDIR 2012). Barcelona, Spain (2012)
8. Erdős, P., Rényi, A.: On the evolution of random graphs. Magyar Tud. Akad. Mat. Kutató Int. Közl 5, 17–61 (1960)
9. Fruchterman, T., Reingold, E.: Graph drawing bt force-directed placement. Software-Practice & Experience 21(11), 1129–1164 (NOV 1991)
10. Herman, I., Melancon, G., Marshall, M.: Graph visualization and navigation in information visualization: A survey. Visualization and Computer Graphics, IEEE Transactions on 6(1), 24 –43 (jan-mar 2000)
11. Keim, D.: Information visualization and visual data mining. IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS 8(1), 1–8 (JAN-MAR 2002)
12. Rohrdantz, C., Oelke, D., Krstajic, M., Fischer, F.: Real-Time Visualization of Streaming Text Data: Tasks and Challenges. In: Workshop on Interactive Visual Text Analytics for Decision-Making at the IEEE VisWeek 201. Providence, RI, USA (2011)
13. Shen, Z., Ma, K.L., Eliassi-Rad, T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. IEEE Transactions on Visualization and Computer Graphics 12(6), 1427–1439 (Nov 2006), http://dx.doi.org/10.1109/TVCG.2006.107
14. Wong, P.C.: Guest editor's introduction: Visual data mining. IEEE Computer Graphics and Applications 19, 20–21 (1999)