new/s/leak – A Tool for Visual Exploration of Large Text Document Collections in the Journalistic Domain

Kathrin Ballweg, Florian Zouhar, Patrick Wilhelmi-Dworski, Tatiana von Landesberger,*
Uli Fahrer, Alexander Panchenko, Seid Muhie Yimam Chris Biemann,† Michaela Regneri, Heiner Ulrich‡

ABSTRACT

Journalists strive for newsworthy stories for the public. To find those stories they need to explore and read documents from large collections such as the Kissinger Cables. This is very time consuming, since the the text document collections are too large to read them alone – even in a team. Interactive text visualization can support journalists in this endeavor. Several tools exists, but interviews with our collaboration journalists revealed their various drawbacks. Therefore, we develop and present a prototype of our novel system *new/s/leak*, which combines natural language processing and visualization adapted specifically to the journalists' needs.

Index Terms: H.5.2 [User Interfaces]: Natural language—; H.5.2 [User Interfaces]: User-centered Design—

1 INTRODUCTION

Journalists wish to find newsworthy stories in large document collections such as the Kissinger Cables or the Panama Papers. To find the stories, journalists need to read the documents in the collection. This is very time demanding and cumbersome due to the collection size. We conducted an in-depth analysis of user needs. It showed that they need tools that help to quickly identify newsworthy information. Journalists need a tool that is easy to learn and use, while providing a wide variety of functions.

Interactive text visualization can support journalists [9, 11, 14]. Our Interviews with journalists showed that available systems [2, 4, 5, 7, 8, 13] cannot cope with large document collections or are too difficult to use and understand.

We have developed a prototype system *new/s/leak* that addresses journalist's needs. It combines natural language processing (NLP) and standard interactive visualization in one tool. NLP extracts important metadata such as entities (people, places etc.) and their relationships. The visualization allows the user to browse the documents according to the extracted information and to read interesting documents. Moreover, it provides interactive data curation functions (e. g., merging two falsely parsed entities) and supports investigation by browsing history.

The tool prototype is a preliminary result of a cooperation project between language technology and visual analytics experts at TU Darmstadt and SPIEGEL Verlag (a large German publisher). The prototype can be accessed on http://newsleakoverview.igd.fraunhofer.de:9000, using Chrome browser. The software code and instructions are available on GitHub under https://github.com/tudarmstadt-lt/newsleak-frontend.

2 RELATED WORK

Text visualization is a broad area within visual analytics. The available approaches are summarized in surveys [9, 11, 14]. We focus on tools for data journalism analyzing large text collections. *DocumentCloud* [7] features archive creation for investigation-related documents. *Overview* [2, 13] is closely related to our tool, however it focuses on document clustering and cluster-based document browsing. *Jigsaw* [5] provides various ways of showing extracted named entities and document data. For data quality, TimeLineCurator [4] offers journalists with visual means to create high quality timeliness from news stories. Two systems are related but not focused on journalism: VaiRoma [3] offers browsing of entities, locations and time in documents. Speculative W@nderverse [8] combines close and distant reading.

The *new/s/leak* system builds upon two systems: *Network of the Day* [1] and *Network of Names* [10]. They extract and visualize named entities and their relations from document collections. *new/s/leak* provides more data processing and visualization features also for exploration history and entity data curation.

3 SYSTEM OVERVIEW

The tool addresses journalistic needs gathered via semi-structured interviews with journalists at SPIEGEL. These revealed that ,most of all, the tools visualizations should be easy to learn and easy to understand. It should support browsing of entities and their relationships over time. The tool should be trustful (i. e. find newsworthy hints in unknown data of which no ground truth exists). For more detail regarding our requirement analysis see http://newsleak.io/2016/02/23/requirements-management/.

The system has two parts: 1) The backend processes input documents to structured data, 2) The frontend shows the data in an interactive interface (see Fig. 2).

3.1 Data and Pre-processing – Backend

The input for the system are raw text documents provided by the journalists. The input data is converted and pre-processed to extract 'dynamic metadata' (e. g., confidentiality, document creation date). Most important for *new/s/leak* are named entities (organization, location, person, miscellaneous) and relationships among them. The Epic named entity tagger [6] is used to detect named entities in documents. Relationships among entities are established based on their co-occurrences in the same document. To enable event-based document exploration, we extract temporal metadata using the Heideltime tool [15]. The backend also stores user-generated data, such as annotations for entities (cf. Fig. 2 – *User Generated Data*) and data curation such as a merged entitites, initially falsely identified by NLP.

3.2 Interactive Visualization – Frontend

The *new/s/leak* interface has 5 linked views used for exploring the data collection from various aspects (cf. Fig. 5): *Frequency Overview*, *Timeline*, *Network View*, *Document View* and *History Tracker*. Frequency, timeline and network views together with free text search can be used to define filters, which determine the list of documents for close reading. User actions are tracked and showed for reproducibility of insights.

The design addresses the low visual literacy and ease of use of journalists by using basic and not task-specialized visualizations (e.g., bar charts, networks, tapped views etc.). View linking and the

*all with GRIS, TU Darmstadt, e-mail:office@gris.tu-darmstadt.de
†all with LT, TU Darmstadt, e-mail:biem@lt.tu-darmstadt.de
‡both with SPIEGEL Verlag, e-mail:heiner.ulrich@spiegel.de

common browsing mechanism (filter add/remove) makes the tool easy to use. Data curation functions address the trustfulness.

Frequency Overview shows the occurrence of entities or of metadata in the selected documents (cf. Fig. 5 – *Frequency Overview*). It uses logarithmic scale reflecting the distribution properties. Blue bars show the frequency in the whole collection, while the black bars show the current filter results (cf. Fig. 9 top).

Timeline shows the frequency of documents over time,again scaled logarithmically. In addition to showing the current filtered frequency, the users can drill down in time to see the document distribution over years, months or days (cf. Fig. 10 – *Interaction*).

Network View shows the document entities as nodes and their relationships as links (cf. Fig. 5). It shows 18 most frequent entities in the filtered document set. Node size denotes the entity frequency in the filtered documents. The node color denotes the entity type. The Plus button uses an entity as a filter.

Annotation and Curation Network view is used also for data curation and annotation. Data curation (e.g., merging or editing entities and their properties) is needed due to quality problems of current NLP algorithms or due to the typos in the original text documents (cf. Fig. 8 bottom). Firstly, the user is able to edit the extracted entities – she can edit their name and their type (e.g., changing London from miscellaneous to location). Secondly, she can merge nodes. This is useful, if the user recognizes that two displayed nodes are actually the same (e.g., USA and United States). The user can hide or delete irrelevant nodes. She can annotate entities for sharing insights with colleagues.

Document View is composed of the *Document Overview* and the *Opened Document View*. The *Document Overview* shows a list of filtered documents with their title or subject (cf. Fig. 5). It shows top 50 documents, while more documents are loaded on demand (cf. Fig. 5 – *Document Overview*). The user can browse the list and open documents for close reading (cf. Fig. 7). The *Opened Document View* shows the document text, where the entities displayed in the graph are underlined (cf. Fig. 5 and Figure 7). The color corresponds to the entity type. The filtered entities are highlighted with background color. This 'close reading' mode enables users to verify hypotheses they generate in the 'distant reading' [12] views.

History Tracker shows the journalists' browsing interactions as meaningful icons (cf. Fig. 5 – *History Tracker*, Figure 6 top). The tracked interactions are: free-text search, filtering by metadata (e. g., classification level) or time and annotating. As the number of interaction may be large during one session, we propose a scalable view. As default, the most important information is displayed: the currently active filters. On demand, the user can see the whole interaction history. The view shows the type and the name of the interaction. More information is provided on demand in a dropdown menu. This menu is also used for removing active filters (cf. Fig. 5 – *History Tracker*, Fig. 6 bottom). Moreover, the user is able to reset filters in the History Tracker (cf. Fig. 6 bottom). This view is useful for review, reproducibility and sharing of analytical paths.

An example of its current usage shows an accompanying video and in Fig. 11. It shows the journalistic research when only having a rough idea what a collection is about (here: Enron Mails).

3.3 Technical Background

Interaction between the backend and the frontend is enabled via a decoupled API (cf. Fig. 4 – *Frontend-Backend-Interface*). The API integration facilitates the independent development of both software components. The API can be directed either to the database or the *ElasticSearch* (cf. Fig. 4 – *Data Model Management*).

Backend: Input data are processed along the pipeline shown in Fig. 3. Metadata and source texts are stored in a *PostgreSQL* database and retrieved by *ElasticSearch*. It offers a fast data access for performance critical operations (e. g. faceted search).

Frontend: The interactive visualization component consists of three units. These are the visualization libraries (D3JS), user interface libraries (AngularJS) and the frontend library management (RequireJS, Bower Package Manager) (see Fig. 2 & 4).

For more detailse see [16]

3.4 Development Process

The tool is developed in close cooperation with SPIEGEL. After initial requirement analysis (see above), tool development started. Monthly meetings gave us feedback on the current state and ideas for further improvements. Moreover, after a first prototype was finished (in Spring 2016), we conducted a small-scale usability study with 10 university students. It showed the needs for usability improvement such as timeline settings, entity highlighting and system responsiveness. These issues were improved and ElasticSearch was included instead of sole DB data management. Our experiences are documented in our blog `http://newsleak.io/`.

The tool is now prototypically available to SPIEGEL journalists for testing and feedback. The final version is expected in January 2017. In the future, we focus on exploring events (i.e., time+place+person), links between documents (e.g., sender-receiver) and linking data to other sources (e.g., pictures, Wikipedia).

ACKNOWLEDGEMENTS

We thank TU Darmstadt students and Franziska Lehmann for their help with the system implementation and the requirement analysis. We thank SPIEGEL Verlag journalists for their requirement analysis and tool feedback. This work has been financed by VW Foundation (Grant No. 90 847.).

REFERENCES

[1] D. Benikova, U. Fahrer, A. Gabriel, M. Kaufmann, S. M. Yimam, T. von Landesberger, and C. Biemann. Network of the day: Aggregating and visualizing entity networks from online sources. In *Proc. NLP4CMC Workshop at KONVENS*, Hildesheim, Germany, 2014.

[2] M. Brehmer, S. Ingram, J. Stray, and T. Munzner. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE TVCG*, 20(12):2271–2280, 2014.

[3] I. Cho, W. Dou, D. X. Wang, E. Sauda, and W. Ribarsky. Vairoma: a visual analytics system for making sense of places, times, and events in roman history. *IEEE TVCG*, 22(1):210–219, 2016.

[4] J. Fulda, M. Brehmel, and T. Munzner. Timelinecurator: Interactive authoring of visual timelines from unstructured text. *IEEE TVCG*, 22(1):300–309, 2016.

[5] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *IEEE TVCG*, 19(10):1646–1663, 2013.

[6] D. Hall, G. Durrett, and D. Klein. Less grammar, more features. In *Proc. of Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pages 228–237. ACL, June 2014.

[7] T. Han, J. Reese, Gradestaff, and A. DeBarros. DocumentCloud.

[8] U. Hinrichs, S. Forlini, and B. Moynihan. Speculative practices: utilizing infovis to explore untapped literary collections. *IEEE TVCG*, 22(1):429–438, 2016.

[9] S. Jänicke, G. Franzini, M. Cheema, and G. Scheuermann. On close and distant reading in digital humanities: A survey and future challenges. *Proc. of EuroVisSTARs*, pages 83–103, 2015.

[10] A. Kochtchi, T. v. Landesberger, and C. Biemann. Networks of names: Visual exploration and semi-automatic tagging of social networks from newspaper articles. *EG CGF*, 33(3):211–220, 2014.

[11] K. Kucher and A. Kerren. Text visualization browser: A visual survey of text visualization techniques. *Poster at IEEE VIS*, 2014.

[12] F. Moretti. *Graphs, maps, trees : abstract models for a literary history*. Verso, London, UK, 2007.

[13] Overview Services Inc. Overview.

[14] A. A. Puretskiy, G. L. Shutt, and M. W. Berry. Survey of text visualization techniques. *Text mining: applications and theory*, pages 105–127, 2010.

[15] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[16] S. M. Yimam, H. Ulrich, T. von Landesberger, M. Rosenbach, M. Regneri, A. Panchenko, F. Lehmann, U. Fahrer, C. Biemann, and K. Ballweg. new/s/leak – information extraction and visualization for investigative data journalists. In *Proceedings of ACL-2016 System Demonstrations*, pages 163–168, Berlin, Germany, August 2016. Association for Computational Linguistics.

Pictures Appendix

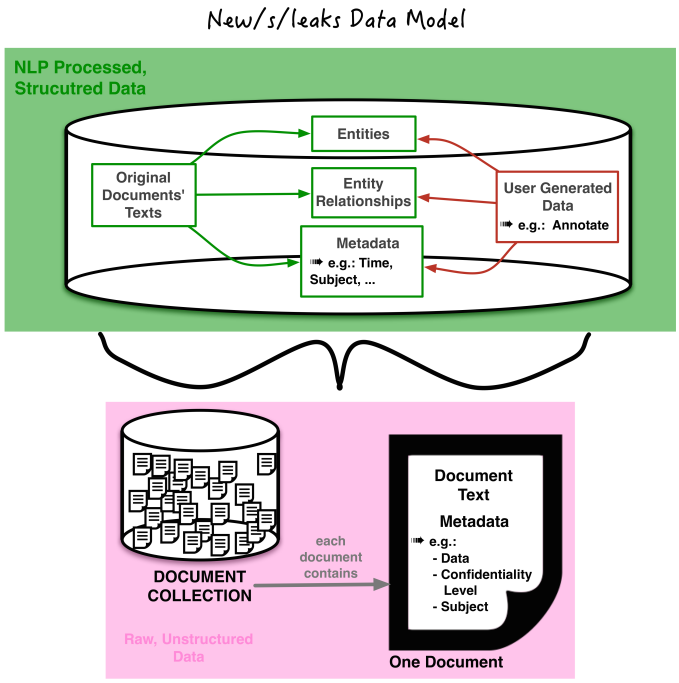


Figure 1: *New/s/leak's* data model and the connection from the raw and unstructured data to the NLP processed, structured data.

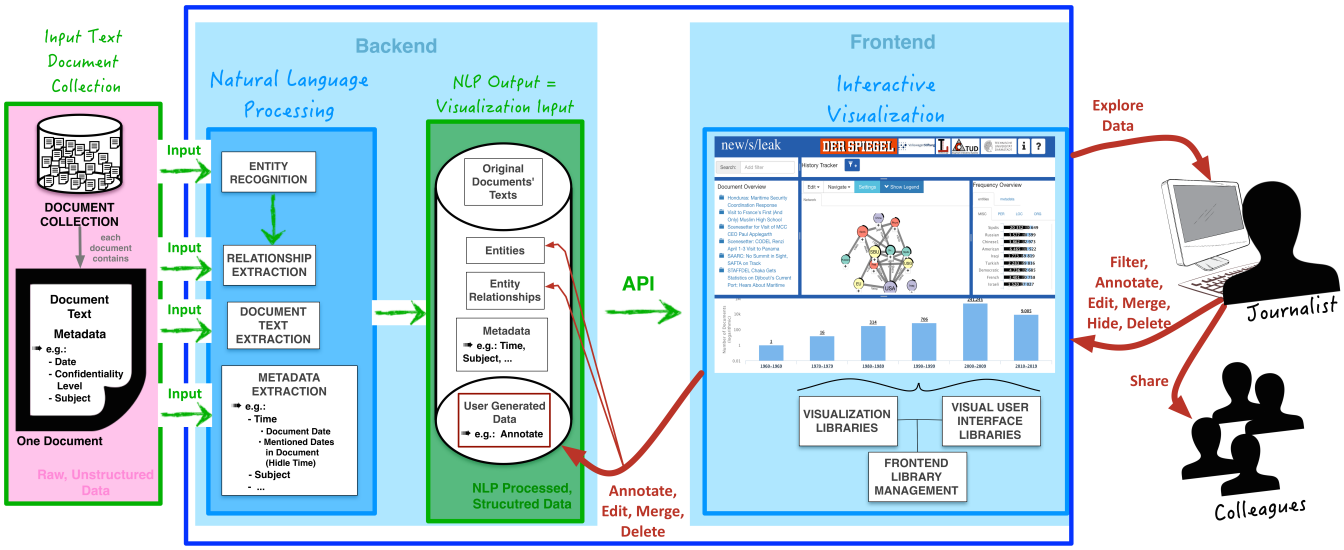
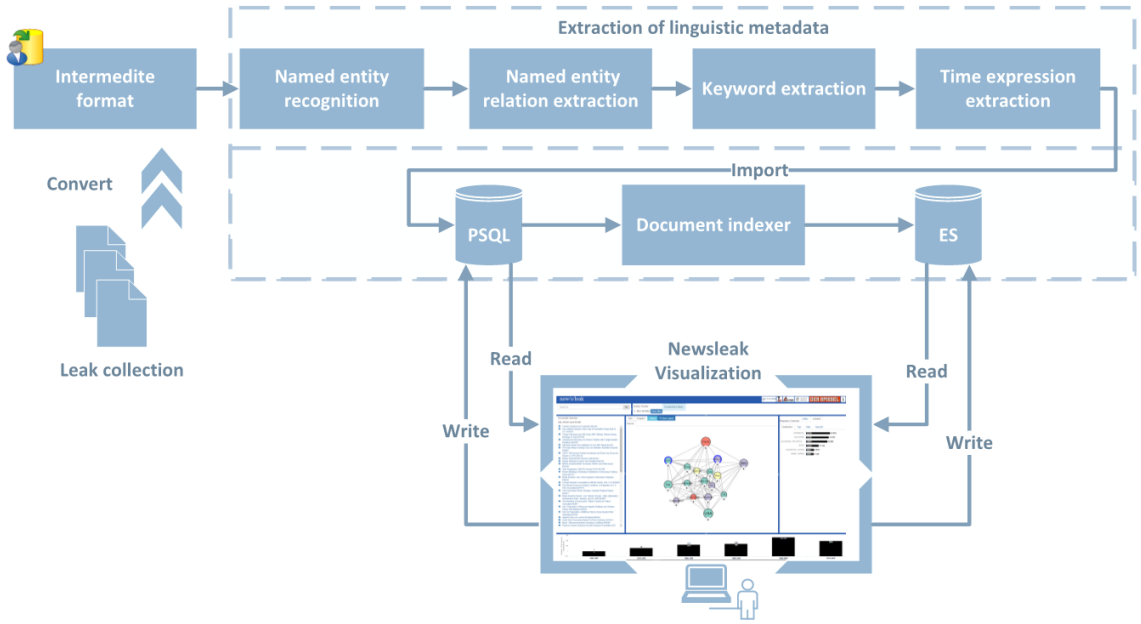
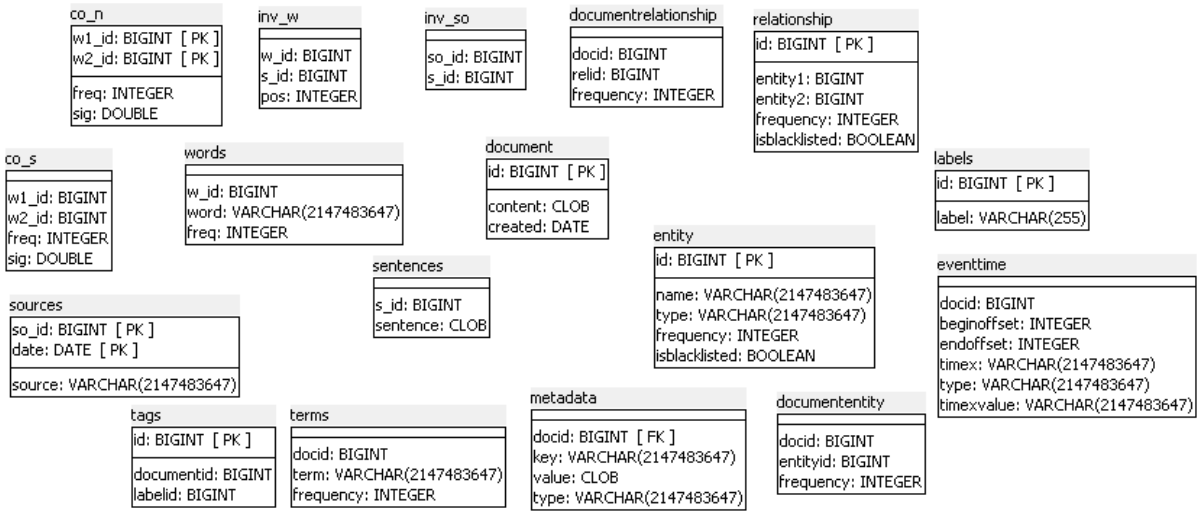


Figure 2: *New/s/leak's* system overview from an entire system perspective – combined with in- and output data



(a) Data processing pipeline: Leak collection contains the raw text collection (e.g. Kissinger Cables). It is converted to *intermediate format*. It contains CSV files that contain the texts, data stamps and further document metadata. This is done as an input to the pipeline for *extracting linguistic metadata*. This pipeline first extracts *named entities*, then their *relations*, *keywords* and *times*. The result is imported to *PSQL* database, *Document indexer* indices the data for *ElasticSearch*.



(b) Database schema

Figure 3: Data processing pipeline (a) and database schema (b) of *new/s/leak*

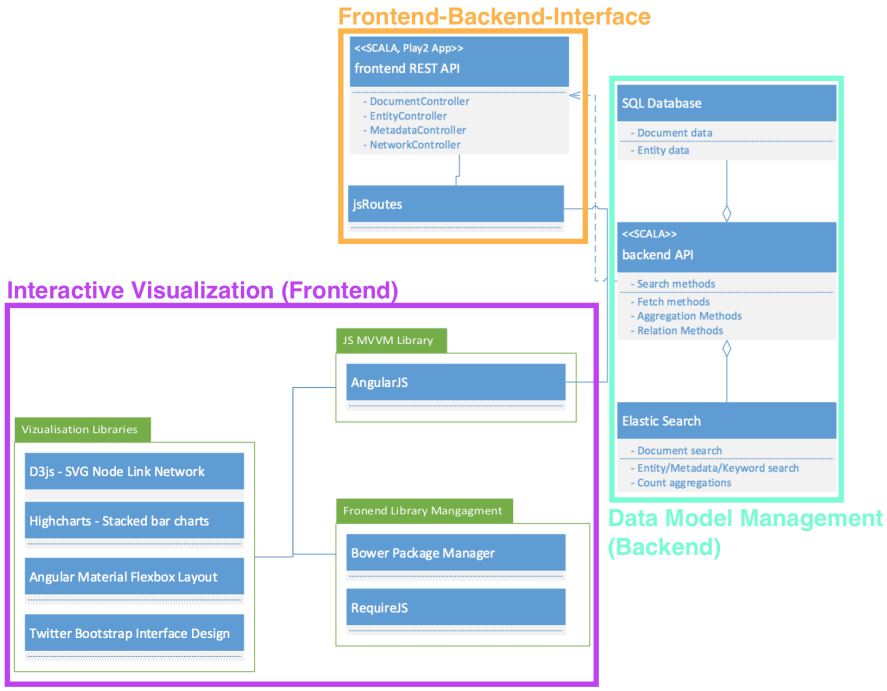


Figure 4: *New/s/leak's* system overview from a technical perspective

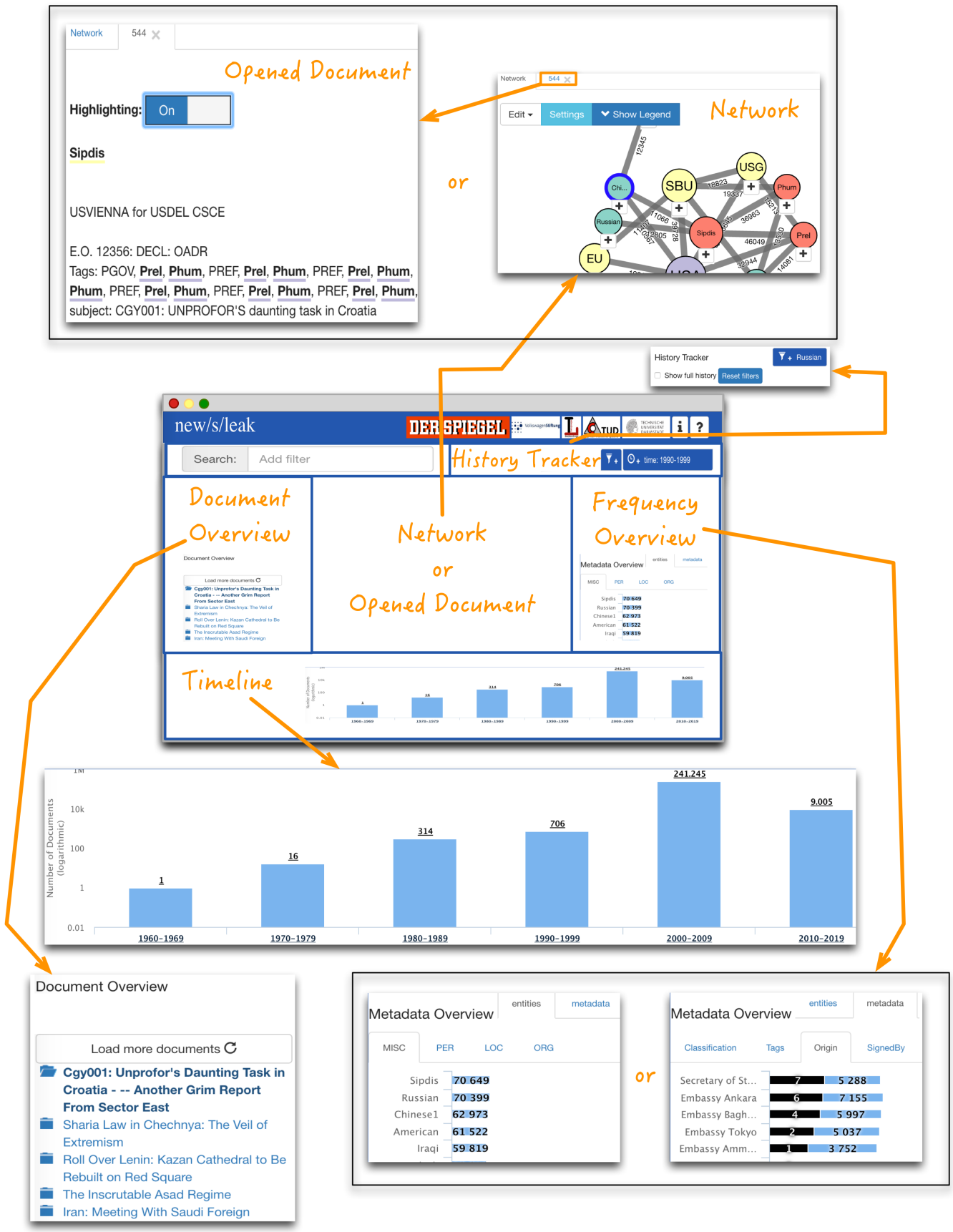


Figure 5: *New/s/leak's* multi-view visual interface

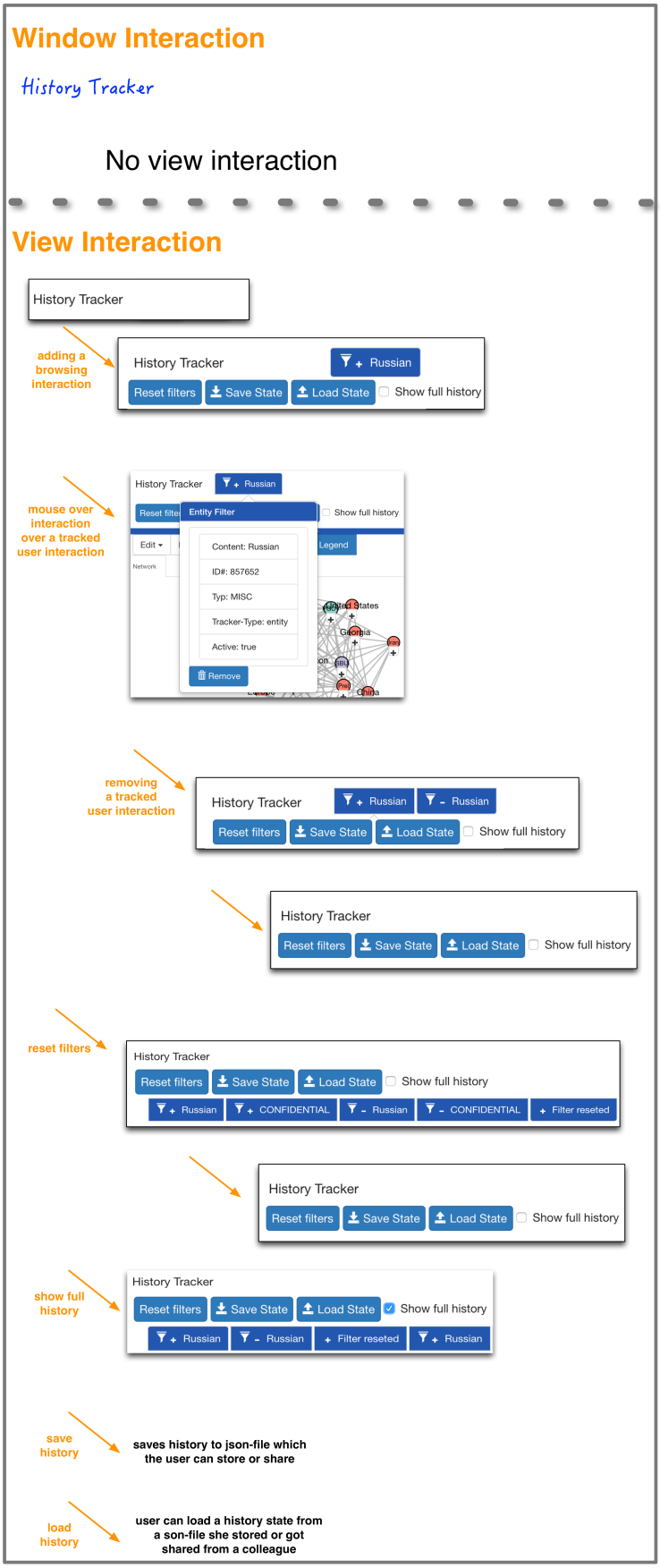


Figure 6: Interactions of the *History Tracker* – grouped by *Window* and *View Interaction*

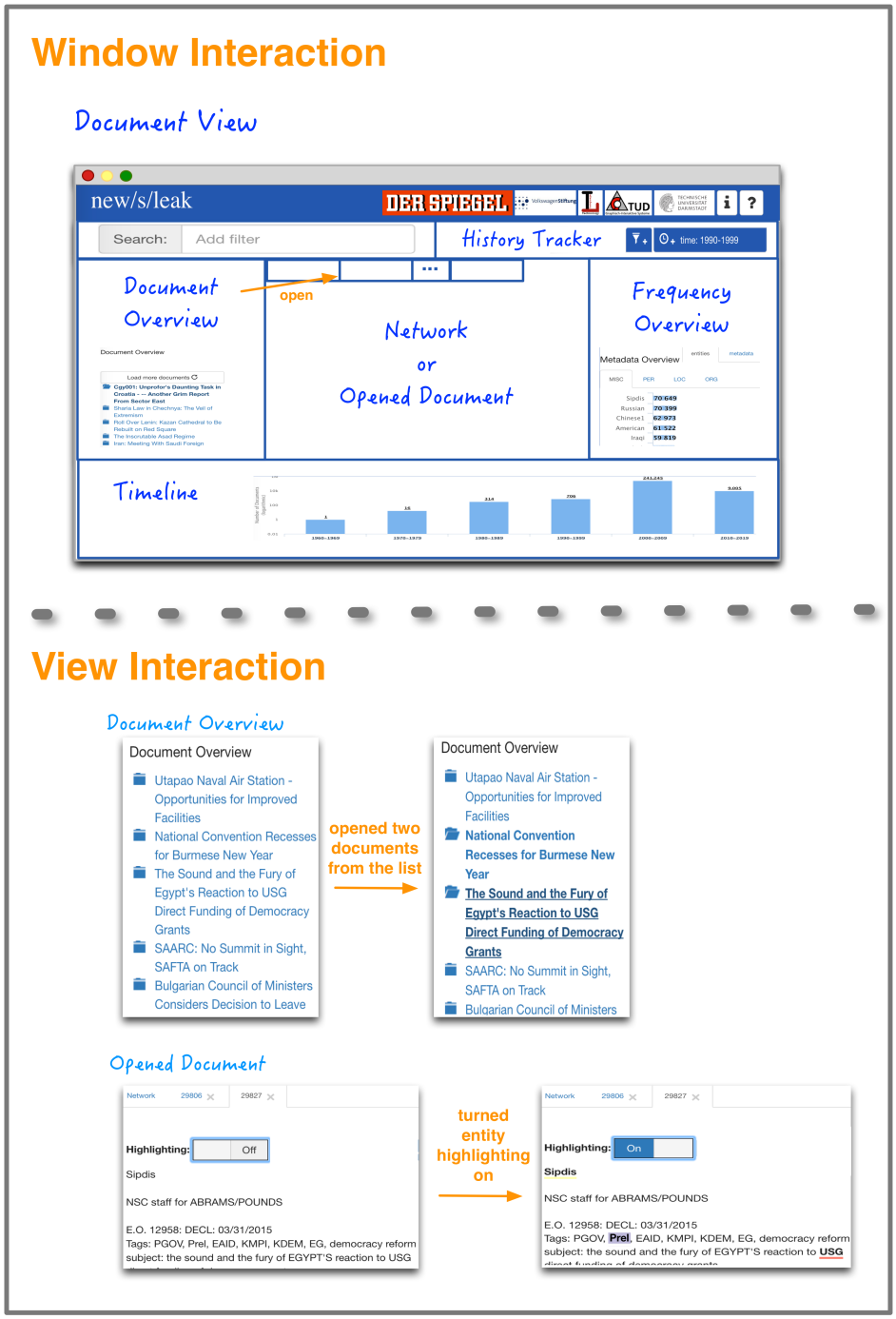


Figure 7: Interactions of the *Document View (Document Overview & Opened Document)* – grouped by *Window* and *View Interaction*

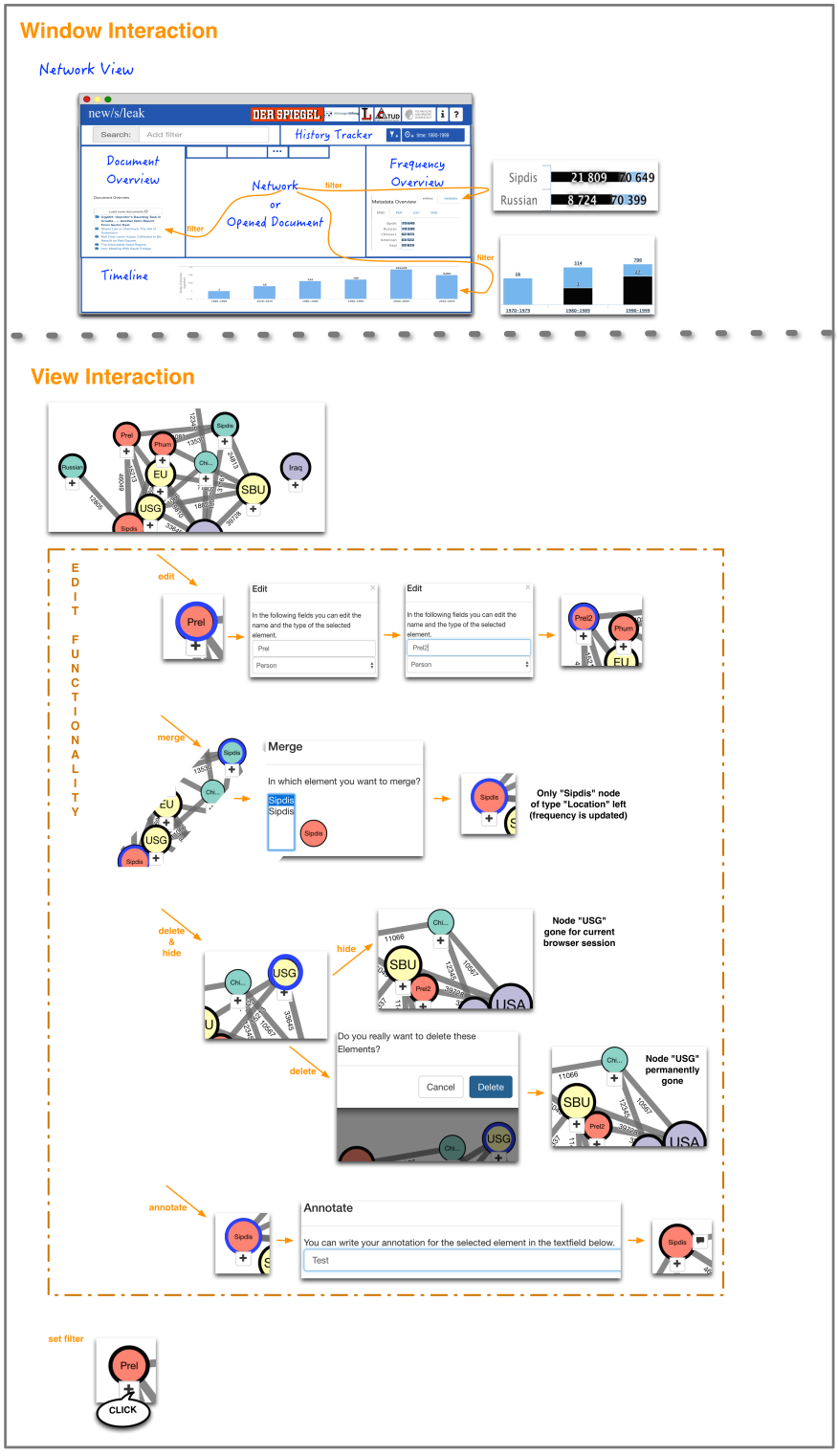


Figure 8: Interactions of the *Network View* – grouped by *Window* and *View Interaction*

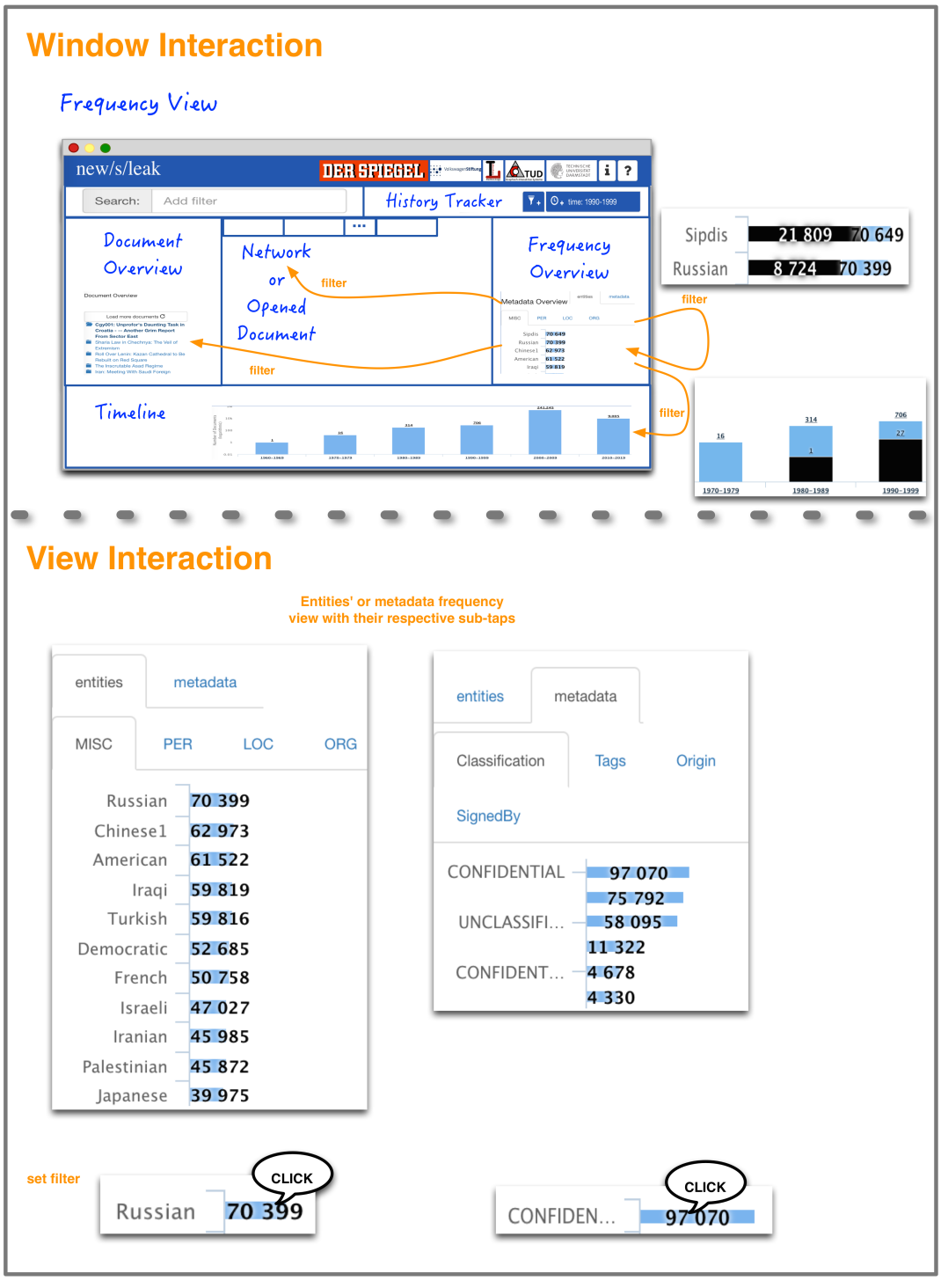


Figure 9: Interactions of the *Frequency View* – grouped by *Window* and *View Interaction*

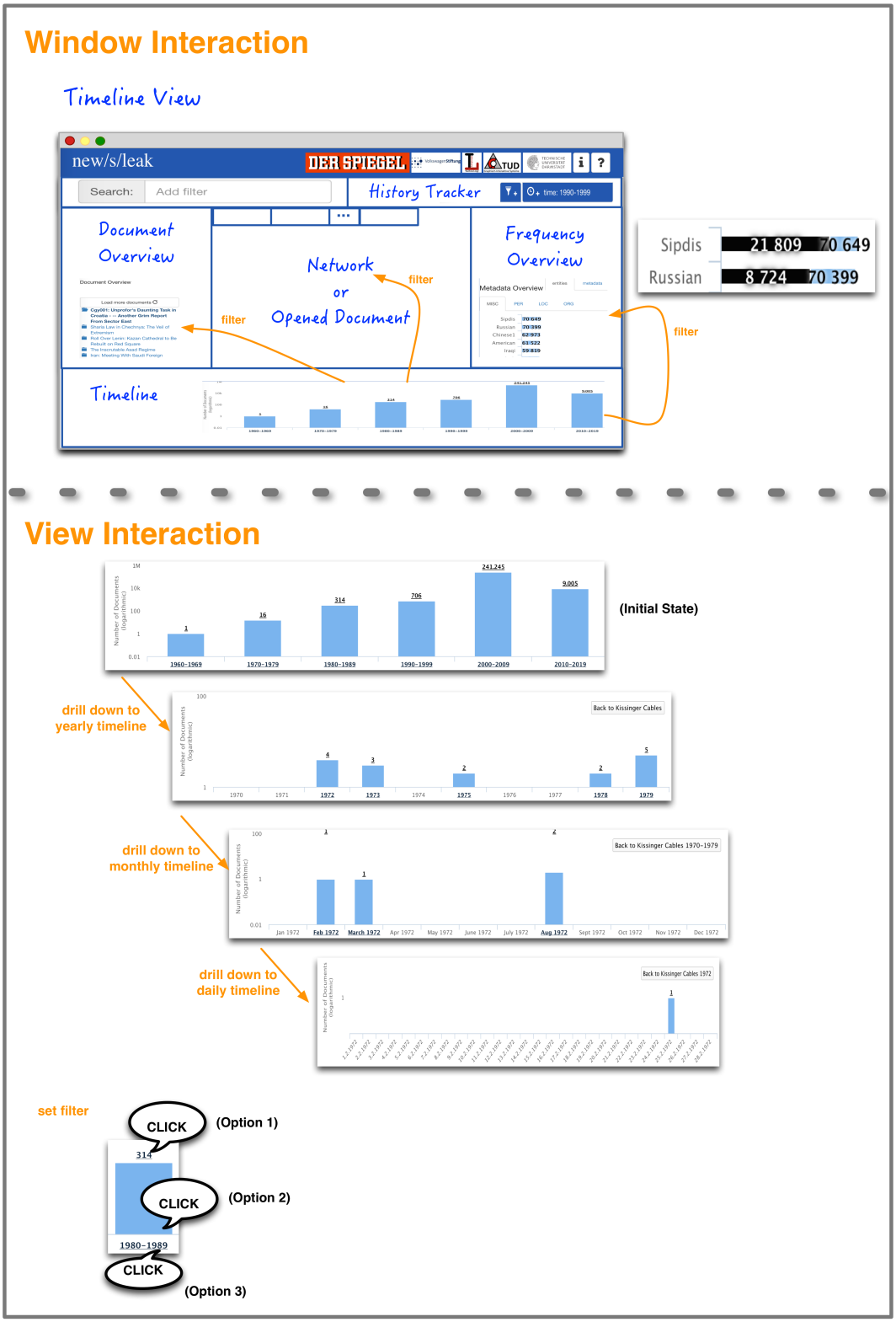


Figure 10: Interactions of the *Timeline View* – grouped by *Window* and *View Interaction*

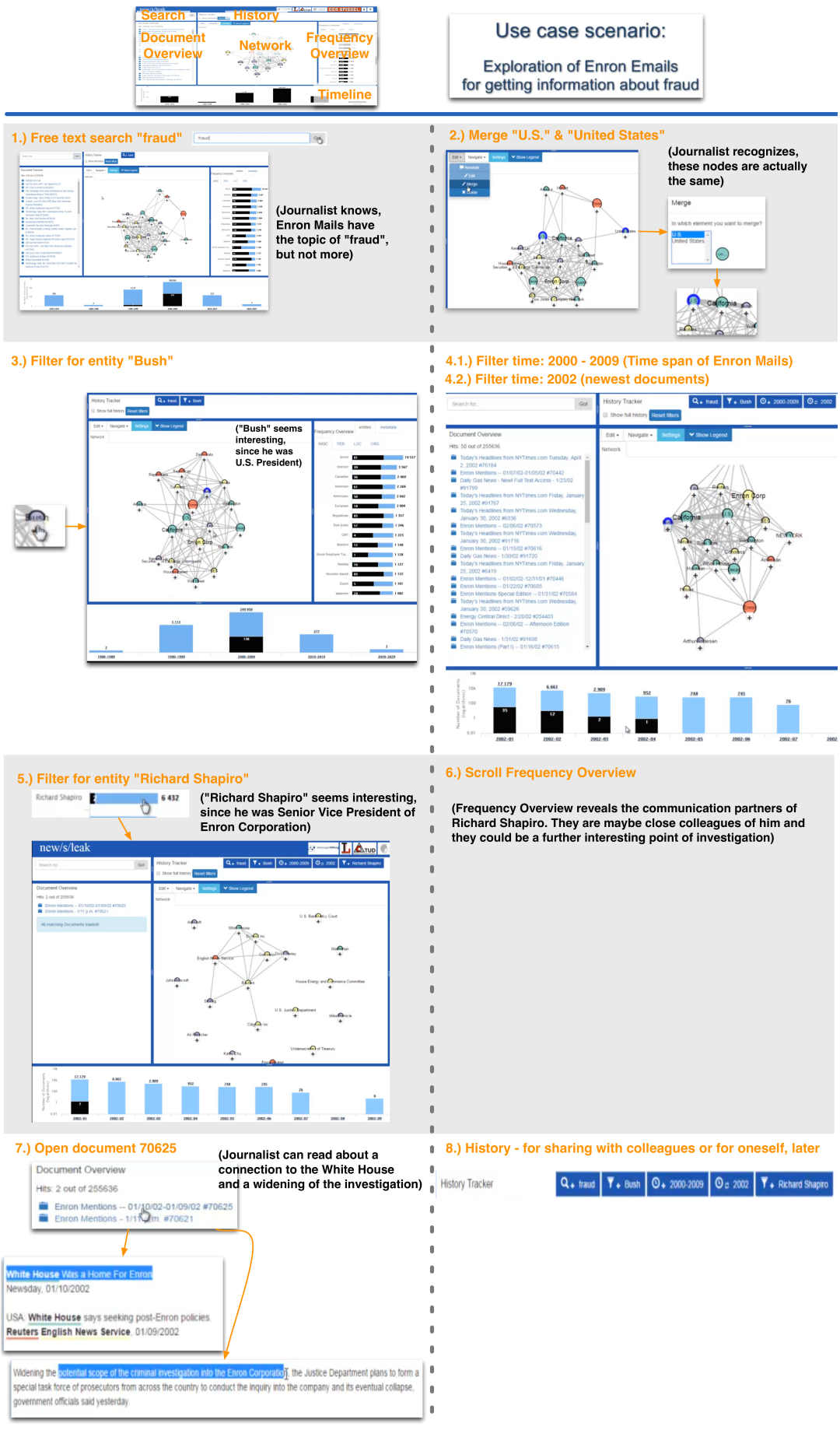


Figure 11: Use case for the usage of *new/s/leak* – journalistic research when only having a rough idea what a document collection is about