new/s/leak – Information Extraction and Visualization for Investigative Data Journalists

Seid Muhie Yimam[†] and **Heiner Ulrich**[‡] and **Tatiana von Landesberger**[◇] and
Marcel Rosenbach[‡] and **Michaela Regneri**[‡] and **Alexander Panchenko**[†] and
Franziska Lehmann[◇] and **Uli Fahrer**[†] and **Chris Biemann**[†] and **Kathrin Ballweg**[◇]

[†]**FG Language Technology**
Computer Science Department
Technische Universität Darmstadt

[◇]**Graphic Interactive Systems Group**
Computer Science Department
Technische Universität Darmstadt

[‡]**SPIEGEL-Verlag**
Hamburg, Germany

Abstract

We present *new/s/leak*, a novel tool developed for and with the help of journalists, which enables the automatic analysis and discovery of newsworthy stories from large textual datasets. We rely on different NLP preprocessing steps such named entity tagging, extraction of time expressions, entity networks, relations and metadata. The system features an intuitive web-based user interface based on network visualization combined with data exploring methods and various search and faceting mechanisms. We report the current state of the software and exemplify it with the WikiLeaks PlusD (Cablegate) data.

1 Introduction

This paper presents *new/s/leak*[1], the *network of searchable leaks*, a journalistic software for investigating and visualizing large textual datasets (see live demo here[2]). Investigation of unstructured document collections is a laborious task: The sheer amount of content can be vast, for instance, the WikiLeaks PlusD[3] dataset contains around 250 thousand cables. Typically, these collections largely consist of unstructured text with additional metadata such as date, location or sender and receiver of messages. The largest part of these documents are irrelevant for journalistic investigations, concealing the crucial storylines. For instance, war crime stories in WikiLeaks were hidden and scattered among hundreds of thousands of routine conversations between officials. Therefore, if journalists do not know in advance what to look for in the document collections, they can only vaguely target all people and organizations (named entities) of public interest.

Currently, the discovery of novel and relevant stories in large data leaks requires many people and a large time budget, both of which are typically not available to journalists: If the documents are confidential like datasets from an informer, only a few selected journalists will have access to the classified data, and those few will have to carry the whole workload. On the other hand, if the documents are publicly available (e.g. a leak posted on the web), the texts have to be analyzed under enormous time pressure, because the journalistic value of each story decreases rapidly if other media publish it before.

There is a plethora of tools (see Section 2) for data journalist that automatically reveal and visualize interesting correlations hidden within large number-centric datasets (Janicke et al., 2015; Kucher and Kerren, 2014). However, these tools provide very limited automatic support with visual guidance through plain text collections. Some tools include shallow natural language processing, but mostly restricted to English. There is no tool that works for multiple languages, handles a large number of text collections, analyzes and visualizes named entities along with the relations between them and allows editing what is considered as an entity. Moreover, available software is usually not open source but rather expensive and often requires substantial training, which is unsuitable for a journalist under time pressure and no prior experience with such software.

[1] http://newsleak.io

[2] http://bev.lt.informatik.tu-darmstadt.de/newsleak/

[3] https://wikileaks.org/plusd/about

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics—System Demonstrations, pages 163–168,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

The goal of *new/s/leak* is to provide journalists with a novel visual interactive data analysis support that combines the latest advances from natural language processing approaches and information visualization. It enables journalists to swiftly process large collections of text documents in order to find interesting pieces of information.

This paper presents the core concepts and architecture behind the *new/s/leak*. We also show an in-depth analysis of user requirements and how we implement and address these. Finally, we discuss our prototype, which will be made available in open source[4] under a lenient license.

2 Related work

Kirkpatrick (2015) states that investigative journalist should look at the facts, identify what is wrong with the situation, uncover the truth, and write a story that places the facts in context. This work explains traditional story discovery engine components which, on the technical side, consist of a knowledge base, an inference engine and an easy to use user interface for visualization.

Discussions with our partner journalists confirm this view: newsworthy stories are not evident from leaked documents, but following up on information from leaks might reveal relevant pointers for journalistic research.

Cohen et al. (2011) outline a vision for "a cloud for the crowd" system to support collaborative investigative journalism, in which computational resources support human expertise for efficient and effective investigative journalism.

The *DocumentCloud*[5] and the *Overview*[6] project are the most popular tools comparable to the *new/s/leak*, both designed for journalists dealing with large set of documents. *DocumentCloud* is a tool for building a document archive for the material related to an investigation. Similarly to our system, people, places and organizations are recognized in documents. We put additional focus on the UI by adding a graph-based visualization and better support for document browsing. *Overview* is designed to help journalists find stories in large number of documents by topical clustering. This is a complementary approach to ours: rather than keyword-based topics, our tool centers around entities and text-extracted relations. Further, we added advanced editing capabilities for users to add entities and edit the whole network.

Aleph.grano.cc visualizes connections of people, places and organizations extracted from text documents, but leaves the relationship types underspecified. *Detective.io*, a platform for collaborative network analysis, does some of the visualizations and annotations we are working on. However, there is a crucial difference: *Detective.io* assumes a rigid data structure (e.g."corporate networks") and the user has to fill the underlying database entirely manually. Our tool targets on unstructured sources and extracts networks from text, enabling navigation to the sources. *Jigsaw*[7] extracts named entities from text collections and computes various plots from them, but received little attention from journalists due to its unintuitive user interface.

With *new/s/leak*, we want to develop existing tools a step further, by combining the automation of entity and relationship extraction with an intuitive and appealing visual interface.

New/s/leak is based on two prior systems developed namely from the works of (Benikova et al., 2014), *Network of the Day*[8] and (Kochtchi et al., 2014), *Network of Names*[9]. Both tools automatically analyze named entities and their relationships and present an interactive network visualization that allows to retrieve the original sources for displayed relations. In the current project, we add support for faceted data browsing, a timeline, a better access to source documents and a possibility to tag and edit visualizations.

3 Objectives and User Requirements

The objective of *new/s/leak* is to support investigative data journalists in finding important facts and stories in large unstructured text collections. The two key elements are an easy-to-use interactive visualization and linguistic preprocessing.

To gain a more precise focus for our development, we conducted structured interviews with potential users. They seek for an answer to the question "Who does what to whom?" – possibly amended with "When and where?". At the same time, they always need access to the source documents, to verify the machine-given answers. The

[4] `http://github.com/tudarmstadt-lt/newsleak`

[5] `https://www.documentcloud.org/home`

[6] `https://https://www.overviewdocs.com`

[7] `http://www.cc.gatech.edu/gvu/ii/jigsaw`

[8] `http://tagesnetzwerk.de`

[9] `http://maggie.lt.informatik.tu-darmstadt.de/thesis/master/NetworksOfNames`

outcome of these interviews are summarized in the following requirements.
1) Identify key persons, places and organizations.
2) Browse the collection, identify interesting documents and read documents in detail.
3) Analyze the connections between entities.
4) Assess temporal evolution of the documents.
5) Explore geographic distribution of events.
6) Annotate documents with findings as well as edit the data to enhance data quality.
7) Save and share selected documents.

While some of these requirements (like document browsing) are standard search features, others (like annotation and sharing) are usually not yet integrated in journalism tools. The final version of our system will include all of these features.

4 Implementation details

4.1 System

The *new/s/leak* tool consists of two major parts shown in Figure 1: The backend provides various NLP tools for document pre-processing and document analysis (Sec. 5), interactive visualization for investigative journalism (Sec. 6). The implementation of the backend and frontend components of *new/s/leak* are integrated in a modular way that allows e.g. adding a different relation extraction mechanism or support for multiple languages.

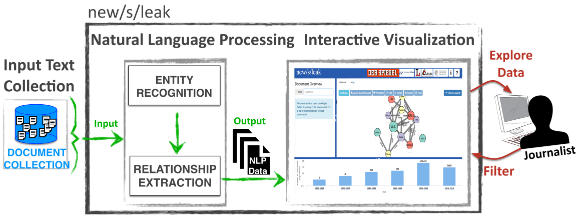


Figure 1: Schema of *new/s/leak* for visual support of investigative analysis in text collections.

4.2 Demo datasets

We demonstrate capabilities of our system on two well-known cases:
1) The well-investigated WikiLeaks PlusD “Cablegate” collection, a collection of diplomatic notes from over 45 years originating from US embassies all over the world.
2) The Enron email dataset (Klimt and Yang, 2004) is a collection of email messages which is available publicly from the Enron Corporation. The dataset comprises over 600,000 messages between 158 employees. This example shows the tools’ general applicability to email leaks.

5 Backend: information extraction

The *new/s/leak* backend uses different NLP tools for preprocessing and analysis, integrated with a relational database, a NoSQL document store and some specific retrieval methods. First, documents are pre-processed and converted to an intermediate *new/s/leak* document representation. This format is generic enough to represent collections from any possible source, such as emails, relational databases or XML documents.

In a second step, our NLP preprocessing extracts important entities, metadata (like time and location), term co-occurrences, keywords, and relationships among entities. We store relevant documents and extracted entities in a *PostgreSQL*[10] database and an *ElasticSearch*[11] index. We explain the following steps using the Wikileaks Cablegate dataset as an example, but the tool is not limited to this single dataset.

5.1 Preprocessing

The first step of *new/s/leak* workflow is preprocessing the input documents and represent them in *new/s/leak*’s intermediate document representation. Preprocessing of the Cablegate dataset requires truecasing the original documents which are mostly in capitals. The work of Lita et al. (2003) indicates that truceasing improves the F-score of named entity recognition by 26%. We used a frequency based approach for case restoring based on a very large true-cased background corpus. Once case is restored, we extract metadata from the document, including the document creation dates, subject, message sender document creator, and other metadata already marked in the dataset. Metadata is stored in a database as triples of (n, t, v) where n is the name, t is the data type (e.g. text or date) and v is the value of the metadata. The tool further supports manual identification of metadata during the analysis and production stages.

[10] `http://www.postgresql.org`
[11] `https://www.elastic.co`

Dataset	#Documents	#Entities	#Relations
WikiLeaks	251,287	1,363,500	163,138,000
Enron	255,636	613,225	81,309,499

Table 1: Statistics on WikiLeaks PlusD and Enron

5.2 Entity, relation, co-occurrence, keyword, and event time extractions

Recognition of named entities and related terms are key steps in the investigative data-driven journalistic process. For this purpose, we first automatically identify four classes of entities, namely person (PER), organization (ORG), location (LOC) and miscellaneous (MISC) using the named entity recognition tool from Epic[12]. We assume relationships between entities whenever the two entities co-occur in a document. In order to extract relevant keywords regarding two entities, we follow the approach by Biemann et al. (2007). Furthermore, we extract entity relation labels by computing document keywords using JTopia[13]. JTopia extracts relevant key terms for search based on part of speech information. To label the relationship between two entities, the most frequent keywords from the documents where the two entities appeared together is used. To extract temporal expressions, we use the Heideltime (Strötgen, 2015) tool, which disambiguates temporal expressions based on document creation times. For the WikiLeaks dataset, it is possible to extract and disambiguate more than 3.9 million temporal expressions.

All the extraction and processing work-flows of the *new/s/leak* components are implemented using the Apache Spark cluster computing framework for parallel computations. Table 1 shows the different statistics for the WikiLeaks PlusD and Enron datasets.

6 Frontend: interactive visualization

Journalists can browse through the document collection using an interactive visual interface (see Figure 2). It enables faceted document exploration within several views:
1) Graph view: shows named entities and their relations.
2) Map view: shows document distribution in geographic space.
3) Document timeline view: shows document frequency over time.
4) Document view: is composed of a) document list and b) document text for reading.

The views are interactive so that the user can browse and explore the document collection on demand. The user starts with exploring entities and their connections in the graph view or by searching for entities and keywords. All interactions in the views define a filter that constrains the current document set, which in turn changes information displayed in the views. The user can assess document frequency in a map or in the timeline, drill down and select documents from the result list, and read them closely. User-selected entities are highlighted in the documents.

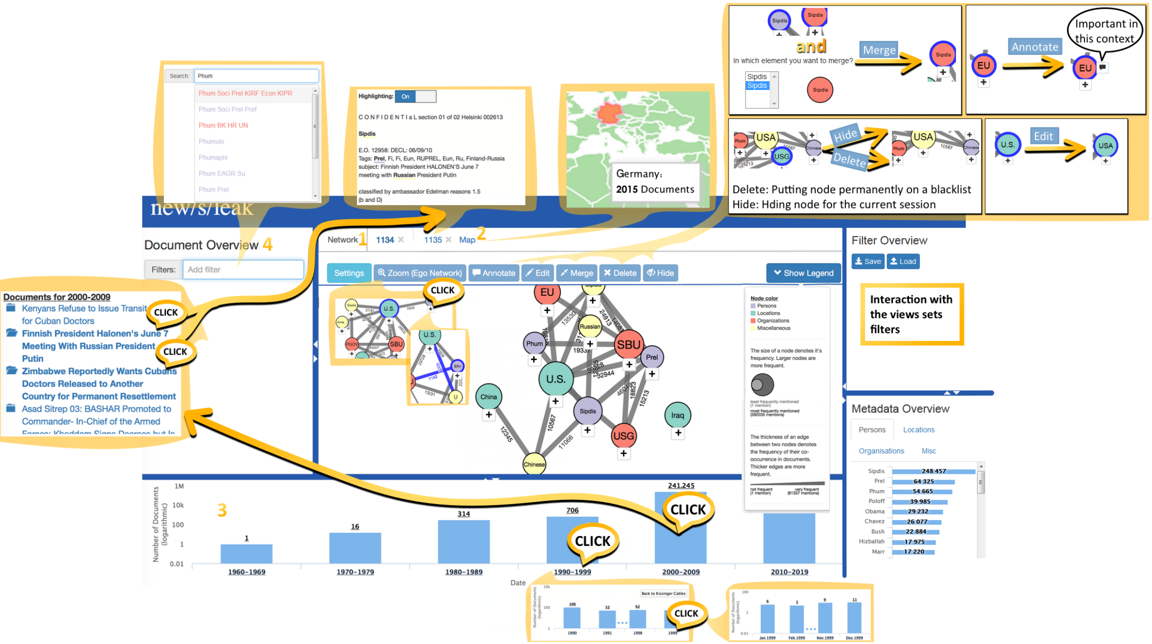
Graph view: entities and their co-occurrences The graph view shows a set of entities as nodes and their connections as links. Node size denotes the frequency of an entity in the document collection, node color denotes the entity type. The co-occurrence of entities within the documents is shown by edge thickness and edge label.
The user can explore the entities and their connections via expanding the graph along the neighbors of a selected entity (plus button). The expanded entities are slowly faded in, so that the user can easily spot which entities appeared. Moreover, the user can drill down into the data by displaying the so-called *ego network* of a selected entity. Clicking on nodes and edges retrieves the respective documents.

Map view. The map view shows the document frequency distribution over the geographic space. Users can hover over a country to see the number of documents mentioning this geographic area, effectively using the map as faceted search.

Document timeline. The document timeline shows the number of documents over time. We use a bar chart with logarithmic scale as it better adheres to the exponential document distribution characteristics. The users can drill down in time to see the document distribution over years, months or days. It is also possible to select a time interval for which the corresponding documents are shown in the document view (see below).

Document view. The document view shows a list of documents with their title or subject as selected by the currently active filters. For large document collections, the documents are loaded

[12] https://github.com/dlwh/epic
[13] http://github.com/srijiths/jtopia



In which element you want to merge?
and
Merge
Annotate
Important in this context
Hide
Delete
Edit
Delete: Putting node permanently on a blacklist
Hide: Hding node for the current session
Germany:
2015 Documents
Highlighting: On
news/s/leak
Document Overview 4
Filters: Add filter
Network 1 1134 × 1135 × Map 2
Settings
Zoom (Ego Network)
Annotate
Edit
Merge
Delete
Hide
Show Legend
Filter Overview
Save
Load
Interaction with the views sets filters
Metadata Overview
Persons
Locations
Organisations
Misc
Documents for 2000-2009
Kenyans Refuse to Issue Transit for Cuban Doctors
Finnish President Halonen's June 7 Meeting With Russian President Putin
Zimbabwe Reportedly Wants Cubans Doctors Released to Another Country for Permanent Resettlement
Asad Sitrep 03: BASHAR Promoted to Commander- In-Chief of the Armed
CLICK
EU
U.S.
SBU
Phum
Russian
Prel
China
Sipdis
USG
Iraq
Chinese
3
Number of Documents (logarithmic)
1M
10k
100
1
0.01
1
16
314
706
241,245
1960–1969
1970–1979
1980–1999
1990–1999
2000–2009
2010–2019
Date
Sipdis 248 457
Prel 64 325
Phum 54 665
Pgov 39 985
Obama 29 232
Chavez 26 077
Bush 22 884
Hizballah 17 975
Marr 17 220

Figure 2: Interactive Visualization Interface composed of graph view on entities, document distribution over time, list of selected documents and a map of document geo-distribution.

on demand. The user can browse the list and identify documents for close reading (bold, open folder icon). The document text view shows the full text of the document, where the entities displayed in the graph are underlined. The underline color corresponds to the type of entity. If the user selected entities in the graph, these entities are highlighted with the background color of the entity. This "close reading" mode enables users to verify hypotheses they perceive in the "distant reading" (Moretti, 2007) visualizations.

6.1 Faceted search

The different levels of visualizations enable users to explore the dataset via *navigation* through document collection (Miller and Remington, 2004). Another highly complementary paradigm for information retrieval used in our system, and also one highly expected by users, is *searching*. During navigation, users explore the document collection interactively, browse with the help of the visualization interfaces, zoom in and out, gradually discovering topics, entities and facts mentioned in the corpus.

In contrast, searching in our system assumes that a user issues a conventional *faceted search* (Tunkelang, 2009) query being a free combination of a full text queries with filters on document metadata. The result of such a faceted search is a set of documents that satisfy the specified facets, i.e. search conditions, formulated at query time, acting as a filter. Here facets can be any metadata initially associated with the document (e.g. date, sender or destination) extracted automatically (e.g. named entities or topics). Search results are presented to the user as a list of documents or in a form of a graph that corresponds to the document view of our system (see Figure 2).

Search and navigation compliment each other during a journalistic investigation. For instance, a user can get an idea about information available in the collection via interactive visualization interface (see Section 6). This navigation session may trigger a concrete journalistic hypothesis. In this case, during the next step a user may want to issue a specific search query to find documents that support or falsify the hypothesis. To implement this search functionality, we rely on *ElasticSearch*.

7 User Study

We conducted a user study to analyze the usability of the provided functions in the interactive visual interface. We asked 10 volunteer students of various major subjects to perform investigative tasks using our tool. The tasks covered all views. They included finding documents of a particular date, opening a document, selecting countries in a map, showing and expanding entity network, assessing entity type etc. We assessed subjective user experience.

The results showed that the implemented visualizations were intuitive and the interactive functions were easy to use for the requested tasks. The volunteers especially appreciated the drill down in the timeline, the fading in of newly appeared nodes and edges in the graph. The legend and tooltips were very often used as a guidance in the interface. Upon users' feedback, we extended zooming and panning functions in the graph and included highlighting of open documents in the document list. We improved the graph layout and look for reducing edge and node overplotting.

8 Conclusion and future work

In this paper, we presented *new/s/leak*, an investigative tool to support data journalists in extracting important storylines from large text documents. The backend of *new/s/leak* comprises of different NLP tools for preprocessing, analyzing and extracting objects such as named entities, relationships, term co-occurrences, keywords and event time expressions. In the frontend, *new/s/leak* provides network visualization with different views supporting navigating, annotating, and editing extracted information. We also developed a demo system presenting the current state of the *new/s/leak* tool and conducted a user study to evaluate the effectiveness of the system.

We are currently extending the tool to meet the remaining journalists' requirements. In particular, we include features for annotating entities and their relations with explanations, for saving a particular view to share it with colleagues or for later use. Moreover, we will provide journalists with the possibility to further edit this sharable view. Additional features for manual data curation will enhance data quality for analysis while ensuring protection of sources and compliance with legal issues. We will also integrate adaptive annotation machine learning approach (Yimam et al., 2016) into *new/s/leak* to automatically identify interesting objects based on the journalists' interaction and feedback. Further, we will investigate pulling in other information from linked open data and the web.

Acknowledgements

The authors are grateful to data journalists at Spiegel Verlag for their helpful insights into journalistic work and for the identification of tool requirements. The authors wish to thank Lukas Raymann, Patrick Mell, Bettina Johanna Ballin, Nils Christopher Boeschen, Patrick Wilhelmi-Dworski and Florian Zouhar for their help with system implementation and conduction of the user study. The work is being funded by Volkswagen Foundation under Grant Nr. 90 847.

References

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

C. Biemann, G. Heyer, U. Quasthoff, and M. Richter. 2007. The Leipzig Corpora Collection – monolingual corpora of standard size. In *Proc. Corpus Linguistics*, Birmingham, UK.

S. Cohen, C. Li, J. Yang, and C. Yu. 2011. Computational journalism: A call to arms to database researchers. In *Proc. CIDR-11)*, pages 148–151, Asilomar, CA, USA.

S. Janicke, G. Franzini, M. F. Cheema, and G. Scheuermann. 2015. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. In *Proc. EuroVis*, Cagliari, Italy.

K. Kirkpatrick. 2015. Putting the data science into journalism. *Commun. ACM*, 58(5):15–17.

B. Klimt and Y. Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *Proc. ECML 2004*, pages 217–226, Pisa, Italy.

A. Kochtchi, T. von Landesberger, and C. Biemann. 2014. Networks of names: Visual exploration and semi-automatic tagging of social networks from newspaper articles. *Computer Graphics Forum*, 33(3):211–220.

K. Kucher and A. Kerren. 2014. Text visualization browser: A visual survey of text visualization techniques. online `textvis.lnu.se`.

L.V. Lita, A. Ittycheriah, S. Roukos, and N. Kambhatla. 2003. tRuEcasIng. In *Proc. ACL '03*, Sapporo, Japan.

C. S. Miller and R. W Remington. 2004. Modeling information navigation: Implications for information architecture. *Human-computer interaction*, 19(3):225–271.

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

J. Strötgen. 2015. *Domain-sensitive Temporal Tagging for Event-centric Information Retrieval*. Ph.D. thesis, University of Heidelberg.

D. Tunkelang. 2009. Faceted search. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–80.

S. Yimam, C. Biemann, L. Majnaric, Š. Šabanović, and A. Holzinger. 2016. An adaptive annotation approach for biomedical entity and relation recognition. *Brain Informatics*, pages 1–12.