Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens

Angelos Christos Anadiotis, Oana Balalau, Francesco Chimienti, Mhd Yamen Haddad, Stéphane Horel, Youssr Youssef, Théo Bouganim, Ioana Manolescu, Helena Galhardas

Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens

A.-C. Anadiotis
École Polytechnique & IPP
France
angelos.anadiotis@polytechnique.edu

O. Balalau, T. Bouganim, F. Chimienti
Inria & IPP
France
name.surname@inria.fr

H. Galhardas
INESC-ID & IST, Univ. Lisboa
Portugal
hig@inesc-id.pt

M.-Y. Haddad
Inria & IPP
France
mhd-yamen.haddad@inria.fr

S. Horel
Le Monde
France
horel@lemonde.fr

I. Manolescu, Y. Youssef
Inria & IPP
France
name.surname@inria.fr

ABSTRACT

Investigative Journalism (IJ, in short) requires **combining highly heterogeneous digital datasets** coming from a wide variety of sources. We have developed ConnectionLens, a system that integrates such sources into a single heterogeneous graph and enables users to query the graph using keywords. The first iteration of the system [7] followed a mediator architecture which severely constrained its query scalability. Thus, we **fully re-engineered the system**, moving it to a warehouse architecture, and replacing its core components (information extraction, data querying, and interactive interfaces), which allowed us to handle uses cases orders of magnitude larger than the previous platform. In a consortium of computer scientists and investigative journalists, we propose to demonstrate ConnectionLens' capability to integrate arbitrary heterogeneous datasets and query them flexibly by means of keywords. Among several scenarios, our main focus will be on a **real-world journalistic use case** about situations which may lead to Conflicts of Interest between biomedical experts and various organizations, such as corporations, lobbies, etc. The demonstration will showcase the end-to-end data analysis pipeline, illustrate each system component, and the different parameters governing graph creation and querying.

CCS CONCEPTS

• **Information systems** → **Graph-based database models**; **Information integration**.

KEYWORDS

investigative journalism, heterogeneous datasets, keyword search

ACM Reference Format:
A.-C. Anadiotis, O. Balalau, T. Bouganim, F. Chimienti, H. Galhardas, M.-Y. Haddad, S. Horel, and I. Manolescu, Y. Youssef. 2021. Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3481982

1 INTRODUCTION

Journalism and the press are critical ingredients of any modern society. Like many other industries, such as trade, or entertainment, journalism has benefitted from the explosion of Web technologies, which enabled instant sharing of their content with the audience. However, unlike trade, where databases and data warehouses had taken over daily operations decades before the Web age, *many newsrooms discovered the Web and social media, long before building robust information systems where journalists could store their information and/or ingest data of interest for them.*

Simultaneously, highly appreciated journalism work often requires *acquiring, curating, and exploiting large amounts of digital data.* Among the authors, S. Horel co-authored the "Monsanto Papers" series, which obtained the European Press Prize Investigative Reporting Award in 2018 [2]; a similar project is the "Panama Papers" (later known as "Offshore Leaks") series of the International Consortium of Investigative Journalists [1]. In such projects, journalists must work with *heterogeneous* data, which they often find in *different data models* (*structured* such as relations, *semistructured* such as JSON or XML documents, or graphs, including but not limited to RDF, as well as *unstructured text*). We, the authors, are currently collaborating on such an **Investigative Journalism** (IJ, in short) application, focused on the study of **situations potentially leading to conflicts of interest**[1] (CoIs, in short) between biomedical experts and various organizations: corporations, industry associations, lobbying organizations, or front groups. Information of interest in this setting comes from: scientific publications (in PDF) where authors declare, e.g., "Dr. X. Y. has received consulting fees from ABC"; semi-structured metadata (typically XML, used for instance in PubMed), where authors may also specify such connections; a medical association, say, French cardiology, may build its disclosure database which may be relational, while a company may disclose its ties to specialists in a spreadsheet.

To address the nature of the data, characterized by a high degree of heterogeneity, in par with the nature of the queries, which require the discovery of connections among entities, we have built ConnectionLens. This system integrates data in a graph model, then queries the graph using keywords to get a set of trees connecting all

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
2021. ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481982

[1] According to the 2011 French transparency law, "A conflict of interest is any situation where a public interest may interfere with a public or private interest, in such a way that the public interest may be, or appear to be, unduly influenced."

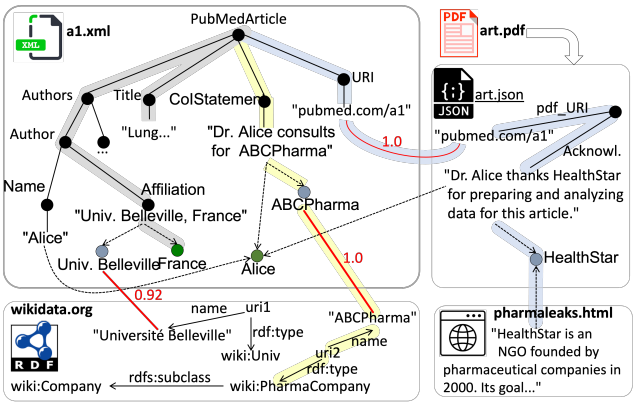

Figure 1: Graph data integration in ConnectionLens.

graph nodes matching the respective keywords. We introduce next the end-to-end data analysis pipeline implemented in ConnectionLens for the CoI IJ application. First, we describe the pipeline, and then we summarize the fundamental parts of our system relevant to the execution of the IJ CoI pipeline. Finally, we explain the demo scenario and outline related work.

2 USE CASE: CONFLICTS OF INTEREST IN THE BIOMEDICAL DOMAIN

The topic. Biomedical experts such as health scientists and researchers in life sciences play an important role in society, advising governments and the public on health issues. They also routinely interact with industry (pharmaceutical, agrifood, etc.), consulting, collaborating on research, or otherwise sharing work and interests. To trust advice from these experts, it is important to ensure vested interests do not unduly influence the advice. However, IJ work, e.g. [15, 16, 21], has shown that disclosure information is often scattered across multiple data sources, hindering access to potential conflicts of interest. We now illustrate the data processing required to gather and collectively exploit such information.

Sample data. Figure 1 shows a tiny fragment of data that can be used to find connections between scientists and companies. *For now, consider only the nodes shown as a black dot or as a text label, and the solid, black edges connecting them; these model directly the data. ConnectionLens add the others as we discuss in Section 3.1.* (*i*) Hundreds of millions of bibliographic notices (in **XML**) are published on the PubMed website; the site also links to research (in PDF). In recent years, PubMed has included an optional CoIStatement element where authors can declare (in free text) their links with industrial players; less than 20% of recent papers use it, and some of those present are empty ("The authors declare no conflict of interest"). (*ii*) Within the **PDF** papers themselves, paragraphs titled, e.g., "Acknowledgments", "Disclosure statement" etc. may contain such information, even if the CoIStatement is absent or empty. This information is accessible if one converts the PDF in a format such as **JSON**. In Figure 1, Alice declares her consulting for ABCPharma in XML, yet the "Acknowledgments" paragraph in her PDF paper mentions HealthStar[2]. (*iii*) A (subset of a) knowledge base (in **RDF**) such as WikiData describes well-known entities, e.g., ABCPharma; however, less-known entities of interest in an IJ scenario are often missing from such KGs, e.g., HealthStar in our example. (*iv*) Specialized data sources, such as a trade catalog or a Wiki website built by other investigative journalists, may provide information on some such actors: in our example, the PharmaLeaks website shows that the industry also funds HealthStar. Such a site, established by a trusted source (or colleague), even if it has little or no structure, is a gold mine to be reused since it saves days or weeks of tedious IJ work. *In this and many IJ scenarios, sources are highly heterogeneous, while time, skills, and resources to curate, clean, or structure the data are not available.*

Sample query. Our application requires *the connections of specialists in lung diseases, working in France, with pharmaceutical companies.* In Figure 1, the edges with *green* highlight and those with *yellow* highlight, together, form an answer connecting Alice to ABCPharma (spanning over the XML and RDF sources); similarly, the edges highlighted in *green* together with those in *blue*, spanning over XML, JSON and HTML, connect her to HealthStar.

The potential impact of a CoI database. A database of known relationships between experts and interested companies, built by integrating heterogeneous data sources, would be a precious asset, allowing, e.g., to select experts without close industrial ties, in a government advisory committee.

3 DATA ANALYSIS PIPELINE

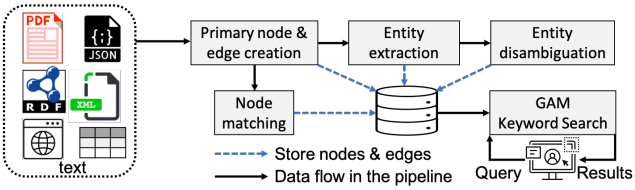

Figure 2: ConnectionLens data analysis pipeline.

The heterogeneous data processing pipeline of our system is outlined in Figure 2. We describe ConnectionLens graph construction, which integrates heterogeneous data into a graph, stored and indexed in a centralized warehouse, in Section 3.1. On this graph, the GAM keyword search algorithm, described in Section 3.2, answers queries such as the one required for our CoI scenario. Technical details as well as experimental validations of choices made in our system are included in [3].

3.1 ConnectionLens graph construction

ConnectionLens integrates JSON, XML, RDF, HTML, relational or text data into a graph, as illustrated in Figure 1. Each source is mapped to the graph as close to its data model as possible, e.g., XML edges have no labels while internal nodes all have names, while JSON conventions are different, etc. Next, ConnectionLens **extracts named entities from all text nodes**, regardless of the data source they come from, using trained language models. In the figure, blue, green, and orange nodes denote Organization, Location, and Person entities, respectively. Each entity node is connected to the text node it has been extracted from by an *extraction edge* recording also the confidence of the extraction (dashed in the figure). For the edges that reflect the structure of the input datasets, we

[2]This example is inspired from prior work of S. Horel where she identified (manually inspecting thousands of documents) an expert supposedly with no industrial ties, yet who authored papers for which companies had supplied and prepared data.

assume confidence 1 and we omit it from the figure. **Entity nodes are shared across the graph**, e.g., Person:Alice has been found in three data sources, Org:BestPharma in two sources etc. ConnectionLens includes a *disambiguation* module which avoids mistakenly unifying entities with the same labels but different meanings. Finally, nodes with similar labels are *compared*, and if their similarity is above a threshold, a **sameAs** (red) edge is introduced connecting them, labeled with the similarity value.

A sameAs edge with similarity 1.0 is called an *equivalence edge*. Then, p equivalent nodes, e.g., the ABCPharma entity and the identical-label RDF literal, would lead to $p(p-1)/2$ equivalence edges. To keep the graph compact, one of the p nodes is declared the *representative* of all p nodes, and instead, we only store the $p-1$ equivalence edges adjacent to the representative. Details on all the above graph construction steps can be found in [3].

Formally, a ConnectionLens graph is denoted $G = (N, E)$, where nodes can be of different types (URIs, XML elements, JSON nodes, etc., but also extracted entities) and edges encode: data source structure, entities extracted from text, and node label similarity.

3.2 The GAM keyword search algorithm

We view our motivating query, on highly heterogeneous content with no a-priori known structure, as a **keyword search query over a graph**. Formally, a query $Q = \{w_1, w_2, \ldots, w_m\}$ is a set of m keywords, and an *answer tree* (AT, in short) is a set t of G edges which (*i*) together, form a tree, and (*ii*) for each w_i, contain at least one node whose label matches w_i. We are interested in *minimal* answer trees, that is answer trees that satisfy the following properties: (*i*) removing an edge from the tree will make it lack at least one keyword match, and (*ii*) if more than one nodes match a query keyword, then all matching nodes are related through sameAs links with similarity 1.0. In the literature (see Section 5), a *score function* is used to compute the quality of an answer, and only the best k ATs are returned for a small integer k. Our problem is harder since: (*i*) our ATs may span over different data sources, even of different data models; (*ii*) they may traverse an edge **in its original or in the opposite direction**, e.g., to go from JSON to XML through Alice; this brings the search space size in $O(2^{|E|})$, where $|E|$ is the number of edges; and (*iii*) **no single score function serves all IJ needs** since, depending on the scenario, journalists may favor different (incompatible) properties of an AT, such as "being characteristic of the dataset" , "being surprising", or on the other hand, "preferring small trees". Thus, **we cannot rely on special properties of the score function**, to help us prune unpromising parts of the search space, as done in prior work (see Section 5). Intuitively, tree size could be used to limit the search: very large answer trees (say, of more than 100 edges) generally do not represent meaningful connections. However, in heterogeneous, complex graphs, users find it hard to set a size limit for the exploration. Nor is a smaller solution always better than a larger one. For instance, an expert and a company may both have "nationality" edges leading to "French" (a solution of 2 edges), but that may be less interesting than finding that the expert has written an article specifying in its CoIStatement funding from the company (which could span over five edges or more).

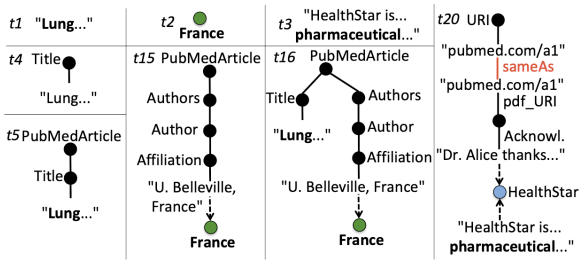

Figure 3: Trees built by GAM for our sample query.

Our **Grow-and-Aggressive-Merge (GAM)** algorithm [3, 5] enumerates trees exhaustively, until a number of answers are found, or a time-out. First, it builds 1-node trees from the nodes of G which match 1 or more keywords, e.g., t_1, t_2, t_3 in Figure 3, showing some partial trees built when answering our sample query. The keyword match in each node label appears in bold. Then, GAM relies on two steps. **Grow** adds to the root of a tree one of its adjacent edges in the graph, leading to a new tree: thus t_4 is obtained by Grow on t_1, t_5 by Grow on t_4, and successive Grow steps lead from t_2 to t_{15}. Similarly, from t_3, successive Grow's go from the HTML to the JSON data source (the HealthStar entity occurs in both), and then to the XML one, building t_{20}. Second, as soon as Grow builds a tree, it is **Merge**d with all the trees already found, rooted in the same node, matching different keywords and having disjoint edges wrt the given tree. For instance, assuming t_{15} is built after t_5, they are immediately merged into the tree t_{16}, having their edges' union. Each Merge result is then merged again with all qualifying trees (thus the "aggressive" in the algorithm name). For instance, when Grow on t_{20} builds a tree rooted in the PubMedArticle node (not shown; call it t_A), Merge(t_{16}, t_A) is immediately built, and is exactly the answer highlighted with green and blue in Figure 1.

Together, Grow and Merge are guaranteed to generate all solutions. If $m = 2$, Grow alone is sufficient, while $m \leq 3$ also requires the Merge step. *GAM may build a tree in several ways*, e.g., the answer above could also be obtained as Merge(Merge(t_{15}, Grow(t_{20})), t_5); GAM keeps a history of the trees already explored, to avoid repeating work on them. Importantly, GAM can be used with any score function. Its details are described in [3, 5] and its scalability with different graph models and sizes is demostrated in [4].

4 SCENARIOS AND USER EXPERIENCE

ConnectionLens is implemented in Java and Python, whereas it relies on PostgreSQL to store the graph. We describe below in detail the CoI scenario at the center of our proposal, then briefly a few other scenarios and the user experience.

The CoI graph. We selected sources based on S. Horel's expertise and suggestions, as follows. (*i*) We loaded **450.000** PubMed bibliographic notices (**XML**), corresponding to articles from 2019 and 2020; they occupy **934 MB** on disk. (*ii*) We have downloaded **42.400** PDF articles corresponding to these notices (those that were available in Open Access), transformed them into **JSON** using an extraction script we developed, and preserved only those paragraphs starting with a set of keywords ("Disclosure", "Competing Interest", "Acknowlegments" etc.) which have been shown [2] to encode potentially interesting participations of people (other than authors) and organizations in an article. Together, these JSON fragments

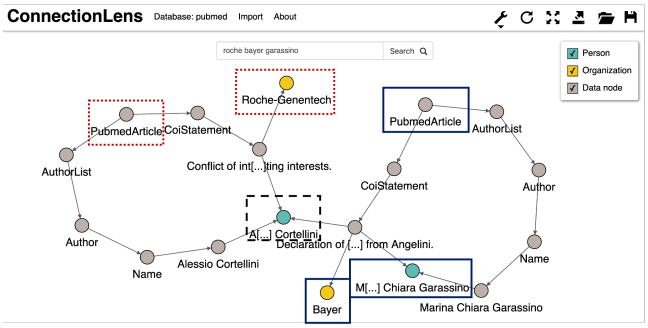

Figure 4: ConnectionLens search screenshot.

occupy **340 MB** on disk. *The JSON and the XML content from the same paper are connected (at least) through the URI of that paper, as shown in Figure 1.* (*iii*) We have crawled 781 **HTML** Web pages from a set of websites describing people and organizations previously involved in scientific expertise on sensitive topics (such as tobacco or endocrine disruptors), specifically: desmogblog.com, tobaccotactics.org, wikicorporates.org and sourcewatch.org. These pages total **24 MB**. *To protect author privacy, we will apply a light pseudonymization on the author names.*

CoI Queries. Our corpus features hundreds of thousands of individuals and organizations. We will prepare a set of 5-10 queries, spanning over the XML, HTML and JSON data sources. A result of the sample 3-keyword query "roche bayer garassino" appears in Figure 4, after some further *interactive result exploration* which allows adding to the visualization panel, *neighbors of query answer nodes*, to grasp more information about them. In the figure, Marina Chiara Garassino, coauthored a paper where she declared a conflict of interest with Bayer, as indicated with the blue, straight-lined boxes. Her coauthor in that paper, Alessio Cortellini (in the black long-dashed-lined box), has published a second paper, where he declared a conflict of interest with Roche, as indicated with the red, short-dashed-lined boxes. Thus, the initial query result connects the three terms. The GUI also allows *adding results of different queries* in the same panel, *eliminating* nodes or edges deemed uninteresting etc. Through gradual query-navigation-graph edit steps, users can identify an interesting part of the data; clicking on each node also shows its original datasource to facilitate verification.

Other scenarios We will also illustrate ConnectionLens on a few other scenarios. For instance, we will use the fact-check corpus available from DataCommons.org (5.500 news fact-checks in JSON, 4.2 MB on disk), a together with a selection of the GDELT global event database (CSV) providing detailed space, time and actor information for each event, and queries returning all fact-checks from a certain organization about a certain actor or event, such as a visit of a president abroad, together with the fact-check authors, event locations etc. Extracted people, organization, and places allow interconnecting the datasets.

Varying system setting ConnectionLens allows controlling: the entity extractor and language used when ingesting the data; the distance functions that control similarity comparisons; the score function used to compare query answers (with different trade-offs between the quality of the keyword matches, the link strength, the answer size, as well as user preferences, injected as a set of keywords, and used to re-weigh answers by their proximity with these keywords). We will vary these and demonstrate the impact on the user experience.

5 RELATED WORK AND CONCLUSION

ConnectionLens was first decribed in [7]. This demonstration reflects a set of changes, outlined below; the interested reader can find more information in the full articles we refer to. First, the system moved from a mediator to a warehouse approach, vastly improving performance [4]. Second, new, better-performing information extraction modules were added, as well as a disambiguation module [4]. Third, the query algorithm has been completely redesigned (see Section 3.2 and [3, 4]).

Our work is clearly a form of *data integration* [9]. The first platform we proposed to Le Monde journalists was a mediator [6], resembling polystores, e.g., [10, 18]. However, we found that: (*i*) their datasets are changing, text-rich and schema-less, (*ii*) running a set of data stores (plus a mediator) was not feasible for them, (*iii*) knowledge of a schema or the capacity to devise integration plan was lacking. Our prior work [7] addressed (*iii*) by introducing keyword search, but it still kept part of the graph *virtual*, and split keyword queries into subqueries sent to sources. Consolidating the graph in a single store, and the centralized GAM algorithm [3] greatly sped up search. We share the goal of exploring and connecting data, with *data discovery* methods [11, 12, 22, 23], which have mostly focused on tabular data.

Keyword search has been studied for XML, e.g., [13, 20], relations, e.g., [24], graphs, e.g., [8, 14], and in particular RDF graphs [19]. However, our problem is harder in several aspects: (*i*) we do not assume any regularity of our text-rich graphs; (*ii*) we allow answer trees to explore edges in both directions; (*iii*) our algorithm is fully orthogonal wrt. the score function, invalidating Dynamic Programming (DP) methods such as [20] and other similar prunings. In particular, we show in [5] that *edges with a confidence lower than 1*, such as similarity and extraction edges in our graphs, compromise, for any "reasonable" score function which reflects these confidences, the *optimal substructure* property at the core of DP. Approximate RDF keyword search algorithms, e.g., [17], are tied to specific score functions, not applicable in our context (many edges are unlabeled, graphs are very heterogeneous etc.)

For a demo preview, see: https://youtu.be/5B0KRow0dv8.

Acknowledgments. The authors thank M. Ferrer and the Décodeurs team (Le Monde) for introducing us, and for many insightful discussions. We also thank Jérémie Feitz and Emmanuel Pietriga for their work on the GUI of ConnectionLens. This work was supported by the ANR AI Chair project SourcesSay Grant no ANR-20-CHIA-0015-01.

REFERENCES

[1] 2013. Offshore Leaks. https://offshoreleaks.icij.org/

[2] 2018. European Press Prize: the Monsanto Papers. https://www.europeanpressprize.com/article/monsanto-papers

[3] Angelos Christos Anadiotis, Oana Balalau, Catarina Conceição, Helena Galhardas, Mhd Yamen Haddad, Ioana Manolescu, Tayeb Merabti, and Jingmao You. 2021. Graph integration of structured, semistructured and unstructured data for data journalism. *Information Systems* (2021), 101846. https://doi.org/10.1016/j.is.2021.101846

[4] Angelos-Christos G. Anadiotis, Oana Balalau, Theo Bouganim, Francesco Chimienti, Helena Galhardas, Mhd Yamen Haddad, Stephane Horel, Ioana Manolescu, and Youssr Youssef. 2021. Empowering Investigative Journalism with Graph-based Heterogeneous Data Management. *IEEE Data Eng. Bull.* accepted for publication (2021).

[5] Angelos-Christos G. Anadiotis, Mhd Yamen Haddad, and Ioana Manolescu. 2020. Graph-based keyword search in heterogeneous data sources. In *Bases de Données Avancés (informal publication)*. arXiv:2009.04283 https://arxiv.org/abs/2009.04283

[6] Raphaël Bonaque, Tien Duc Cao, Bogdan Cautis, François Goasdoué, J. Letelier, Ioana Manolescu, O. Mendoza, S. Ribeiro, Xavier Tannier, and Michaël Thomazo. 2016. Mixed-instance querying: a lightweight integration architecture for data journalism. *PVLDB* 9, 13 (2016). https://doi.org/10.14778/3007263.3007297

[7] Camille Chanial, Rédouane Dziri, Helena Galhardas, Julien Leblay, Minh-Huong Le Nguyen, and Ioana Manolescu. 2018. ConnectionLens: Finding Connections Across Heterogeneous Data Sources. *Proc. VLDB Endow.* 11, 12 (2018), 2030–2033. https://doi.org/10.14778/3229863.3236252

[8] Bolin Ding, Jeffrey Xu Yu, Shan Wang, Lu Qin, Xiao Zhang, and Xuemin Lin. 2007. Finding Top-k Min-Cost Connected Trees in Databases. In *ICDE*. IEEE Computer Society. https://doi.org/10.1109/ICDE.2007.367929

[9] AnHai Doan, Alon Y. Halevy, and Zachary G. Ives. 2012. *Principles of Data Integration*. Morgan Kaufmann. http://research.cs.wisc.edu/dibook/

[10] J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, and S. B. Zdonik. 2015. The BigDAWG Polystore System. *SIGMOD* (2015).

[11] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A Data Discovery System. In *ICDE*. https://doi.org/10.1109/ICDE.2018.00094

[12] Raul Castro Fernandez, Essam Mansour, Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping Semantics: Linking Datasets Using Word Embeddings for Data Discovery. In *ICDE*. https://doi.org/10.1109/ICDE.2018.00093

[13] Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram. 2003. XRANK: Ranked Keyword Search over XML Documents. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, Alon Y. Halevy, Zachary G. Ives, and AnHai Doan (Eds.). ACM, 16–27. https://doi.org/10.1145/872757.872762

[14] Hao He, Haixun Wang, Jun Yang, and Philip S. Yu. 2007. BLINKS: ranked keyword searches on graphs. In *SIGMOD*, Chee Yong Chan, Beng Chin Ooi, and Aoying Zhou (Eds.). https://doi.org/10.1145/1247480.1247516

[15] Stéphane Horel. 2018. *Lobbytomie*. La Découverte. https://www.amazon.fr/Lobbytomie-St%C3%A9phane-HOREL/dp/2707194123/

[16] Stéphane Horel. 2020. Petites ficelles et grandes manoeuvres de l'industrie du tabac pour réhabiliter la nicotine. https://www.lemonde.fr/planete/article/2020/12/19/petites-ficelles-et-grandes-man-uvres-de-l-industrie-du-tabac-pour-rehabiliter-la-nicotine_6063922_3244.html

[17] Gjergji Kasneci, Maya Ramanath, Mauro Sozio, Fabian M. Suchanek, and Gerhard Weikum. 2009. STAR: Steiner-Tree Approximation in Relationship Graphs. In *ICDE*. https://doi.org/10.1109/ICDE.2009.64

[18] Boyan Kolev, Patrick Valduriez, Carlyna Bondiombouy, Ricardo Jiménez-Peris, Raquel Pau, and José Pereira. 2016. CloudMdsQL: querying heterogeneous cloud data stores with a common language. *Distributed Parallel Databases* 34, 4 (2016), 463–503. https://doi.org/10.1007/s10619-015-7185-y

[19] Wangchao Le, Feifei Li, Anastasios Kementsietsidis, and Songyun Duan. 2014. Scalable Keyword Search on Large RDF Data. *TKDE* 26, 11 (2014). https://doi.org/10.1109/TKDE.2014.2302294

[20] Ziyang Liu and Yi Chen. 2007. Identifying meaningful return information for XML keyword search. In *SIGMOD*. https://doi.org/10.1145/1247480.1247518

[21] Naomi Oreskes and Erik Conway. 2012. *Merchants of Doubt*. Bloomsbury Publishing. https://www.amazon.fr/Merchants-Doubt-Scientists-Obscured-Warming/

[22] Masayo Ota, Heiko Mueller, Juliana Freire, and Divesh Srivastava. 2020. Data-Driven Domain Discovery for Structured Datasets. *Proc. VLDB Endow.* 13, 7 (2020). http://www.vldb.org/pvldb/vol13/p953-ota.pdf

[23] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y. Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding related tables. In *SIGMOD*.

[24] Mayssam Sayyadian, Hieu LeKhac, AnHai Doan, and Luis Gravano. 2007. Efficient Keyword Search Across Heterogeneous Relational Databases, Rada Chirkova, Asuman Dogac, M. Tamer Özsu, and Timos K. Sellis (Eds.). IEEE Computer Society, 346–355. https://doi.org/10.1109/ICDE.2007.367880