Towards Efficient Construction of a Traceable, Multimodal, and Heterogeneous Data Warehouse

VLDB 2024 Workshop: VLDB Ph.D. Workshop

Antoine Gauquier

Supervised by: Ioana Manolescu and Pierre Senellart

26/08/2024

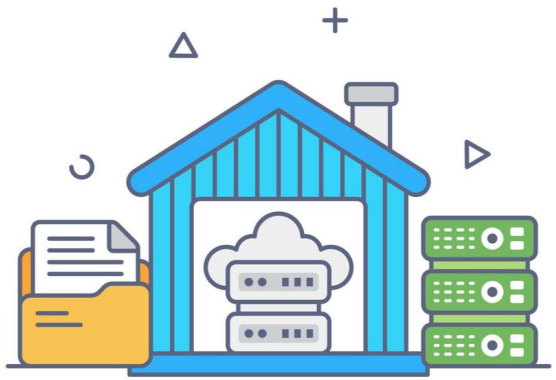
Table of contents

Context

Ever-increasing amount of humanly produced data $\longrightarrow$ countless opportunities for users to find, analyze, and repurpose this data.

Large data volumes and the heterogeneity of formats make these tasks unfeasible by a human $\longrightarrow$ need of automatic methods for gathering, understanding and sharing this data.

This is what we aim for this PhD thesis: efficiently building a traceable, multimodal, and heterogeneous data warehouse.



Organization & Contribution

We divide the subject in three main research axes, each of them relying on modern artificial intelligence methods:

Data Acquisition $\longrightarrow$ Data Extraction & Analysis $\longrightarrow$ Data Warehousing

Main expected contribution: come with a complete pipeline for building the warehouse, where each component benefit from the other ones.

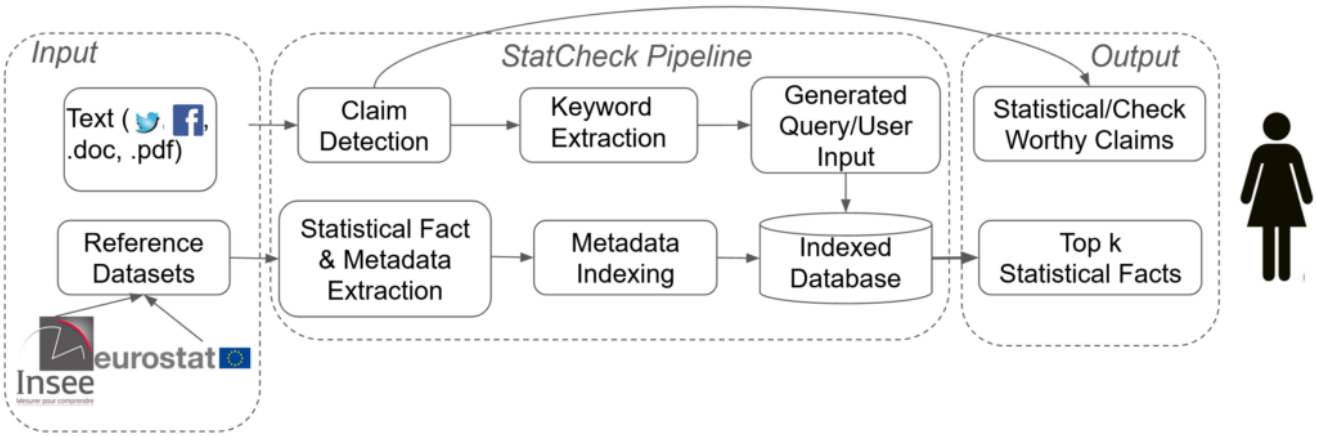
We aim to develop symbiotic relationships between components.

Application: Statistical Data Journalism

Collaboration with journalists from *RadioFrance*.

They are interested in:

- acquiring official statistical data provided by public institutions
- automated tools to verify online claims (e.g., StatCheck [BEG+22])



Example of the StatCheck project, aiming at automatically verifying online claims.

Data Acquisition

The Web shows to be a fruitful choice.

Challenge: Acquiring data files/resources from the Web:

- Massive volumes of data (scalability)...
- ... about a wide range of subjects (semantic diversity)...
- ... formated differently (format heterogeneity)...
- ... while minimizing the effort (efficiency).

Work achieved for this axis presented later.

Data Extraction & Analysis

Data extraction and analysis is made difficult by semantic diversity and format heterogeneity.

Example: Statistical data journalism $\longrightarrow$ covered topics are by definition widely diversified and in a lot of different formats.

Challenge: Come up with a generic method (or methods targeting sets of formats) handling a diversity of formats, independently of the subjects.

Work in progress: generic method for analyzing corpora of multidimensional tabular data (formats: CSV, TSV, XLS, XLSX, etc.).

Data Warehousing

Build a data warehouse platform capable of integrating extracted information.

Challenge: Building a warehouse so that:

- Interconnections are made between common information (or between information sharing common entities).
- We keep track of the information's origins (traceability) $\rightarrow$ to keep track of uncertainty of AI models and explainability of answers.
- Queries can be formulated by non-technical users $\rightarrow$ natural language processing methods.

This axis has not been studied yet.

Table of contents

Problem

Lead to the writing of a research paper (*Efficient Crawler for Scalable Web Data Acquisition*) [GMS], under review.

Problem: Given the starting URL of a website, we want to retrieve as many targets as possible, while minimizing the consumption of resources. We define:

- A target: file uploaded on the website, satisfying some constraints specified by the user (e.g., be a data file: CSV, Excel, PDF...).
- Resources: number of HTTP requests sent to the server, and data volume exchanged with the server.

We want to retrieve the maximum of targets while minimizing the effort to do so.

Research Hypotheses

We rely on two main research hypotheses:

- Hyperlinks similarly structured in the HTML page were they were found lead to similar content (already showed in different contexts in prior works).
- It is possible to learn which hyperlinks are most likely to lead to targets, given their link structure.

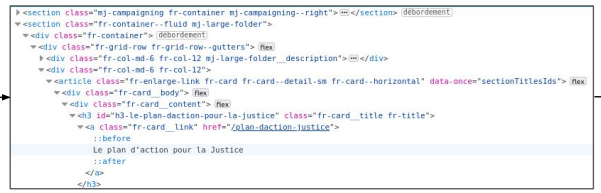
Idea: It is possible to separate, for a given website, parts of it that are rich in targets, from ones that are not (or only a bit), this without any knowledge of the website's structure/content prior to the crawling.

Methodology

How to do it: We represent each encountered hyperlink with its associated DOM path (its link structure).

We form groups of hyperlinks with a merging strategy based on similarity between the DOM paths...
...to separate the fruitful groups from the less interesting ones, as we assume that similar hyperlinks will lead to similar kind of content.

Hyperlink in an HTML page



Associated HTML source code of the hyperlink

```
/html/body.path-frontpage page-node-type-homepage/
[...] section.fr-container--fluid mj-large-folder/
div.fr-container/div.fr-grid-row fr-grid-row--gutters/
div.fr-col-md-6 fr-col-12/article.fr-enlarge-link
fr-card fr-card--detail-sm fr-card--horizontal/
div.fr-card__body/div.fr-card__content/
h3.fr-card__title fr-title/a.fr-card__link
```

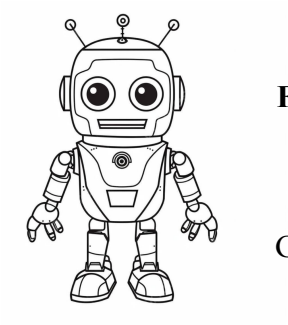
HTML DOM path of the hyperlink

Example of a DOM path found in `https://www.justice.gouv.fr/`

Learning Fruitful Groups: Reinforcement Learning

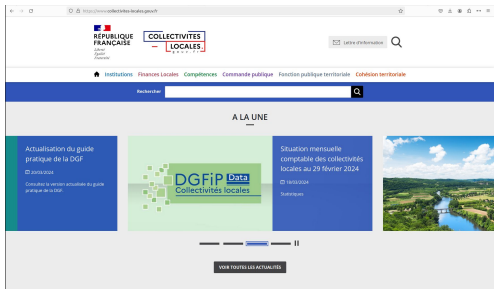
How can we dynamically separate fruitful groups from other ones? By dynamically learning while crawling: Reinforcement learning.

Leveraging Multi-Armed Bandits strategy [ACBF02], and especially Sleeping Multi-Armed Bandits [KNMS10].



Select a group
Follows a hyperlink from said group

Gets a **reward**: a score describing **how "good" was this choice**, to improve future decisions …



How many **targets** we can reach **from this page**?

Reward: Number of reachable, unobserved targets from the HTML page.

URL Classifier

Naive way: Do HTTP HEAD requests over the URL to get Content-Type from the header. But too costly.

Idea: Actively train a URL classifier, taking as input an URL, outputting one of the two following classes: target or HTML.

Doing such an "on–the–fly" training allows the classifier to adapt to its environment and to eventual drastic changes in it.

We do it by taking advantage of URLs that are automatically labeled when following a hyperlink during the crawl.

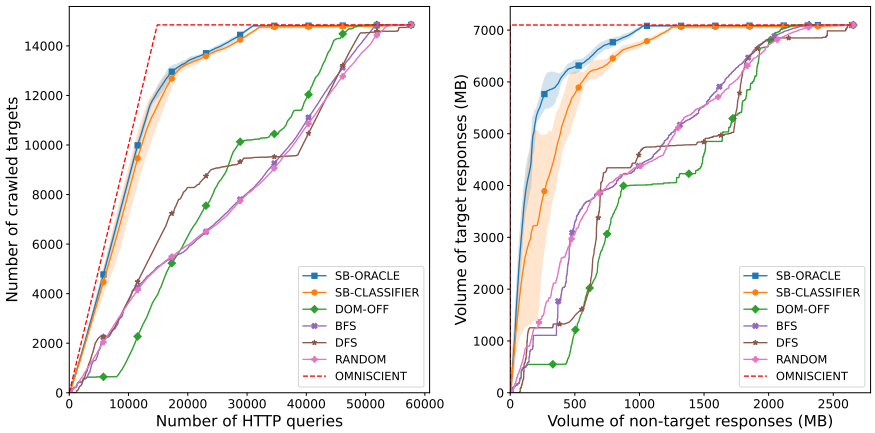
URL Classifier Results (Overview)

Really good results, with some unavoidable errors on Neither class, by design.

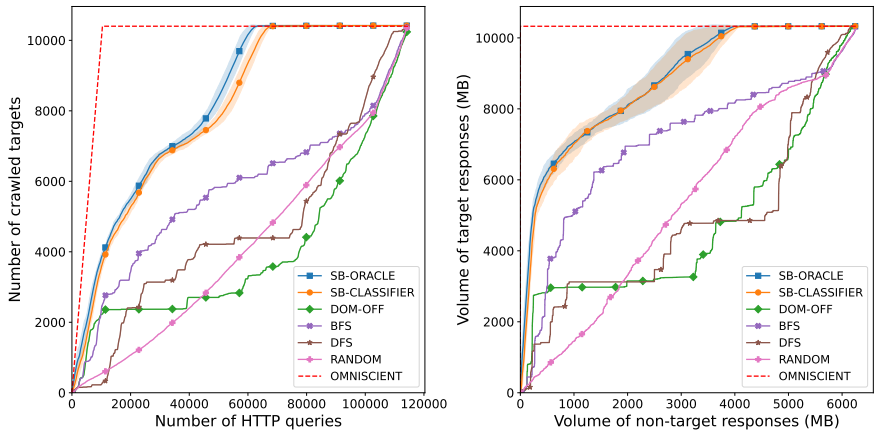
Confusion matrix of our URL classifier, on average over 15 runs, for experimented websites in [GMS] (in percents)

True/Predicted	HTML	Target	Neither
HTML	64.09	0.87	0.00
Target	0.57	26.08	0.00
Neither	7.25	1.74	0.00

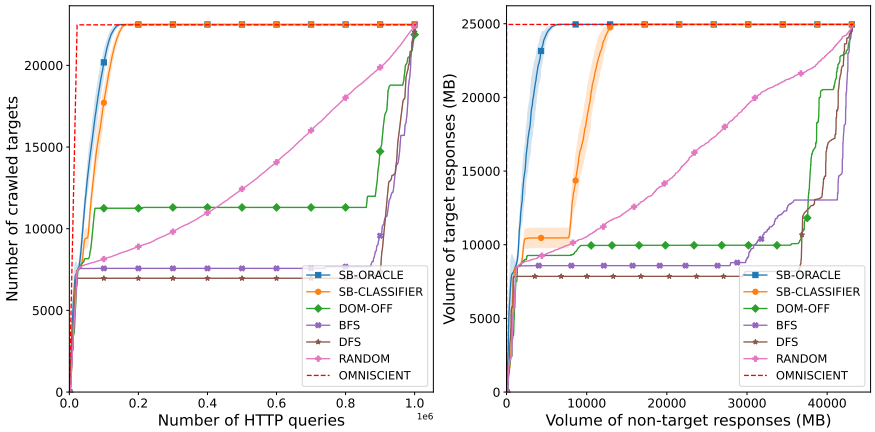
Crawler Results (Overview)



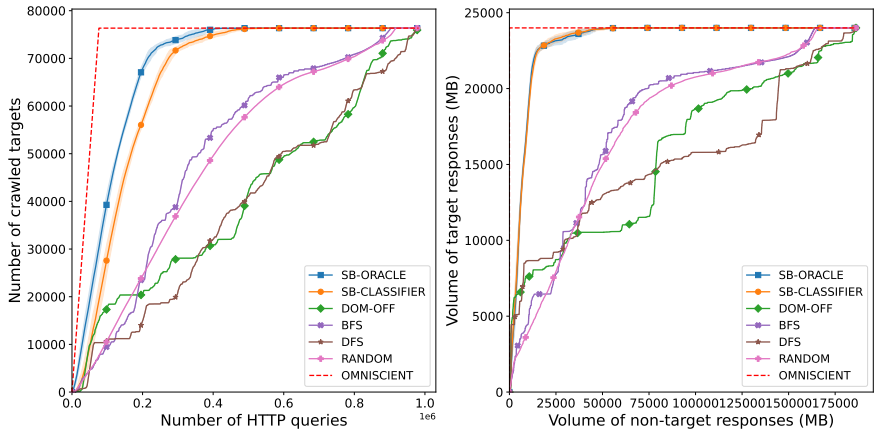
(a) *ju*



(b) *ed*



(c) *in*



(d) *as*

Comparison of different crawler performance for 4 French public institutions

Thank you for your attention!

Any questions?

Bibliography I

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer.

Finite–time Analysis of the Multiarmed Bandit Problem.

Machine Learning, 47(2):235–256, 2002.

Oana Balalau, Simon Ebel, Théo Galizzi, Ioana Manolescu, Quentin Massonnat, Antoine Deiana, Emilie Gautreau, Antoine Krempf, Thomas Pontillon, Gérald Roux, and Joanna Yakin.

Statistical Claim Checking: StatCheck in Action.

In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pages 4798–4802, 2022.

Antoine Gauquier, Ioana Manolescu, and Pierre Senellart.

Efficient Crawler for Scalable Web Data Acquisition.

Under review.

Bibliography II

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma.

Regret Bounds for Sleeping Experts and Bandits.

Machine Learning, 80(2):245–272, 2010.