The "Dataset Creator" Database and Document-Generation Framework

Viewing and Creating Datasets and Custom Dataset Applications

Overview

LTS's new Dataset Creator (dsC) framework can be used to (1) generate open-access data sets for publication and (2) curate corpora of academic books/articles. The dsC framework employs innovative text-encoding and hypergraph data-representation protocols that can enhance the query engines and software platforms driving data-hosting and document-sharing repositories. In particular, this technology enables or improves capabilities such as:

- Ensure that data sets are easy to find from publications, and vice-versa.
- Search data sets and publication texts side-by-side.
- Search data sets according to software/computational metadata — e.g., data-type names, data-field names, procedure signatures, serialization formats, visual display formats, and workflow features.
- Publish data sets with customized built-in "dataset applications" for data visualization/analysis, so that researchers are not required to write their own computer code in order to evaluate and resuse the data set (Figures 1-5, below, exemplify problems with uncurated "raw" data sets).
- Provide machine-readable publication texts with precise representations of individual sentences, paragraphs, keywords/phrases, citations/quotations, identifiers for non-textual content (such as figure illustrations and diagrams), and dataset cross-references.
- Integrate machine-readable text with query results, to support search queries such as: "provide the full text for the first sentence in each document containing the search term" or "return the PDF coordinates of the paragraph-start where a figure illustration is referenced."
- Support integrated document and data query APIs via API client libraries and GUI tools.

Parkinson's Disease Diagnosis: Effect of Autoencoders to Extract Features from Vocal Characteristics

Ashena Gorgan Mohammadi, Pouy

Department of Computer Science, School of Mathem
University of Tehran, 14155-6455, Tehran, Iran, Tel: +

Abstract

The usage of Machine Learning(ML) algorithms in mec
In these methods, the data is first gathered from some
algorithms are applied to predict whether the patient ha
diagnosis in progressive diseases as well. An essential

This article about Parkinson's Disease is discussed here as a case-study to illustrate problems with (1) data sets that publish raw data without accompanying code and (2) publications linked to multiple data sets in a decentralized manner, forcing researchers to expend time aggregating all data into a central location in order to evaluate or reuse the original authors' research.

Figure 1: Parkinson's Article

Figure 2: Parkinson's PhysioNet Dataset

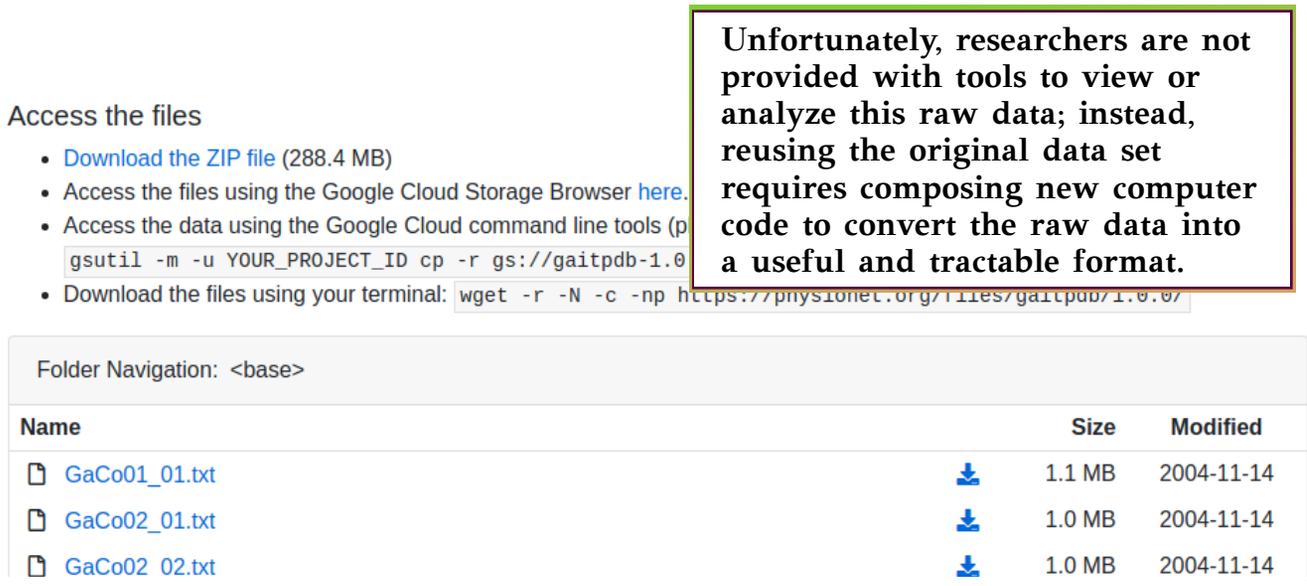

Figure 3: PhysioNet Downloads

Solutions to improve academic text-mining and data-sharing are valuable because publishers have invested in document sharing/indexing platforms (such as SpringerNature, Mendeley, CrossRef, ArXiV, JSTOR, or SciVerse) with the recognition that sophisticated data hosting and document curation capabilities raise the profile of their publications: researchers gravitate toward the most full-featured research platforms, and therefore toward the publications hosted on such platforms.

Indeed, data and document repositories which are full-fledged computational ecosystems — and not merely resource collections — are more likely to become scientists' preferred research tools.

However, contemporary publishing platforms have only rudimentary support for standards and initiatives such as Research Objects and FAIRSHARING (Findable, Accessible, Interoperable, Reusable) or the guidelines advanced by funding sources such as the Bill and Melinda Gates Foundation and

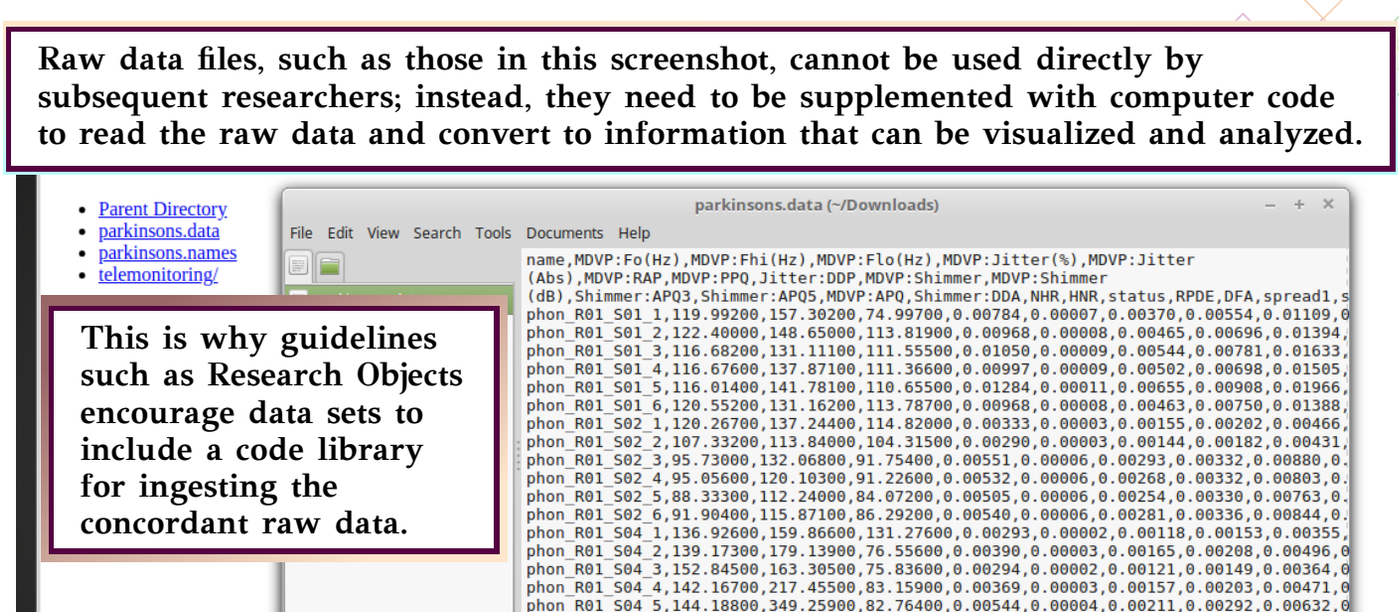

Figure 4: Raw Data is not Useful by Itself

the Chan-Zuckerberg "Invest in Open Infrastructure" program. Tools such as the dsC framework can help publishing platforms realize contemporary standards more effectively, thereby potentially enhancing the prestige, visibility, and popularity of the underlying platform.

The Need for Integrated Data-Hosting and Document Curation

At present, publications (such as academic books or articles) and open-access data sets occupy two largely disconnected ecosystems, where data-hosting services (such as Dryad, Dataverse, OpenScience, or DSpace) are separate from publishers' portals (where books and articles and listed). As a result of such separation, it is impossible to jointly search publications and data sets using a common query engine or **API**. Moreover, it can be a tedious process for researchers to actually locate and examine data sets analyzed in scientific literature.

Typical problems arising from data sets being disconnected from publications are demonstrated in Figure 5 (a case-study derived from an article on how background noise affects perceptions of speech clarity — Jan Holub, *et. al.*, "Subjective Speech Quality Measurement Repeatability: Comparison of laboratory test results," *International Journal of Speech Technology*, 2017); and in Figures 6-9, which show screenshots of an article studying speech, gait, and bioimage markers for Parkinson's disease (under review for the International Journal of Speech Technology). As this Parkinson's article considers different biomarker modalities and therefore studies multiple data sets, it furnishes a good illustration of how the lack of integrated data and publication hosting hinders the research process.

Both of these example articles are linked to multiple data sets in a decentralized fashion, with data distributed across numerous platforms (including dropbox, protocols.io, GitHub, Google Cloud,

PLOSOne, Wikimedia, ArXiV, and Figshare) as well as academic and organizational web sites. In order to reproduce the authors' research, or reuse their methods for new investigations, scientists must retrieve data from multiple sources, in many cases studying the original article's supplemental materials to find secondary literature documenting where the relevant data can be found.

In this speech-technology data set, the raw data (audio) files (1) are hosted on a separate platform from metadata files (2). Also, the raw data file set is published just as a zipped folder, with no process to browse the files other than opening them one at a time. In combination, these factors present time-consuming obstacles to data findability, accessibility, interoperability, and reusability, as defined by the FAIRSHARING initiative.

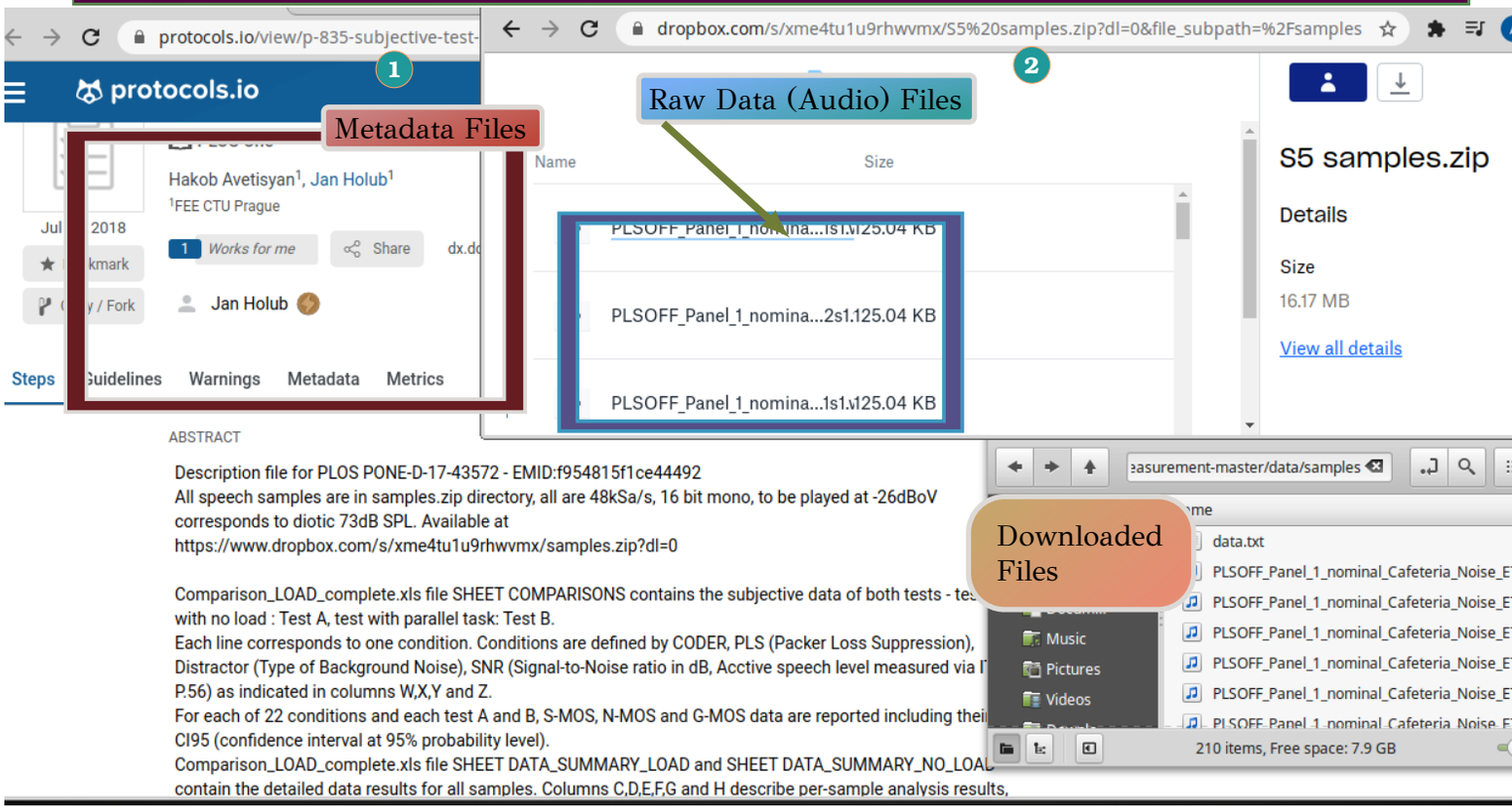

Figure 5: A Case-Study Showing the Limitations of Decentralized Data Hosting

In addition to being a time-consuming bottleneck for research, such ad-hoc data networking makes it difficult for multiple data sets to be merged in a rigorous fashion: this manner of decentralized data-curation leads to data mishmashes with limited scientific integrity, and no reusable digital assets for merging the data into a common format or a "Common Data Model."

Alongside these problem of data decentralization, the majority of published data sets are currently also limited in that they supply only raw data files — without including code to make their raw data accessible to subsequent researchers, either through visual tools or through ingest/analysis algorithms. This forces researchers to write their own code in order to access the data, which can result in unnecessary wasted duplicate efforts, as well as a lack of consistency in how the original data is reused.

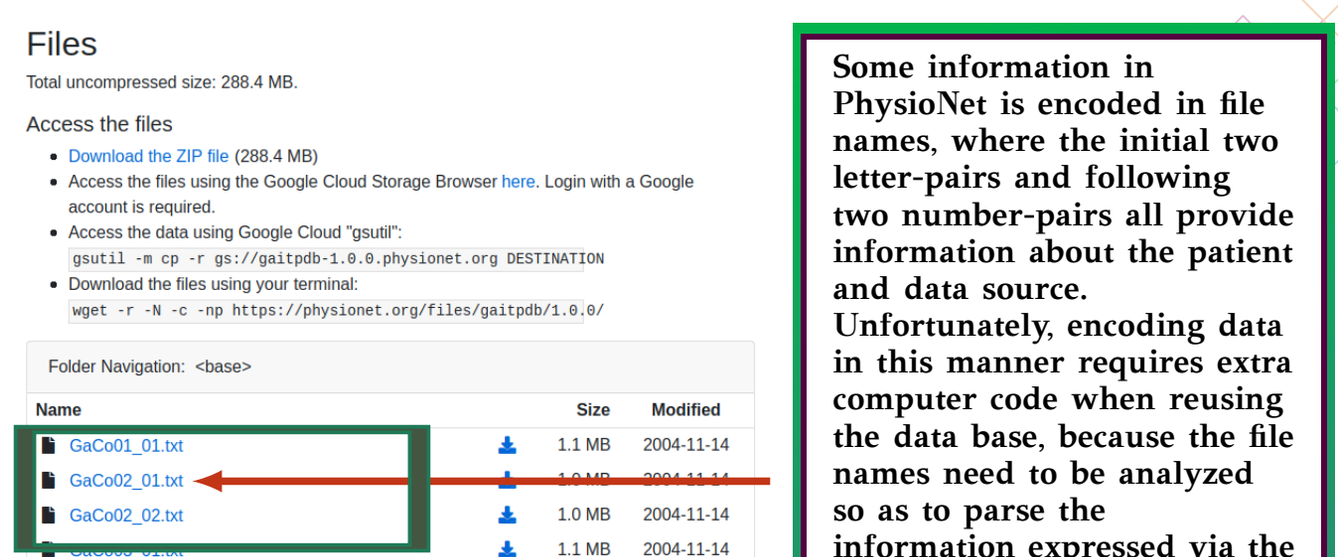

Figure 6: Extracting Information Encoded in File Names

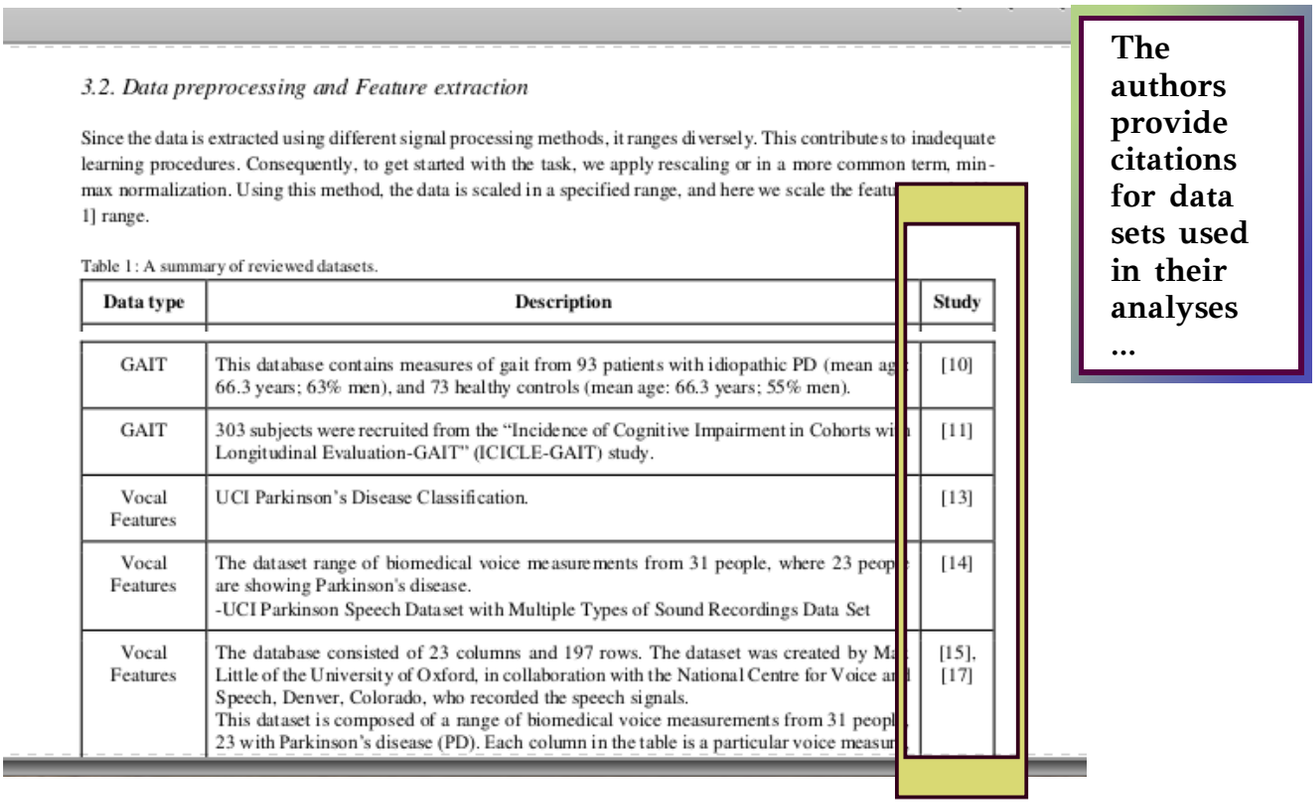

Figure 7: Table Listing Analyzed Data Sets in the Parkinson's Article

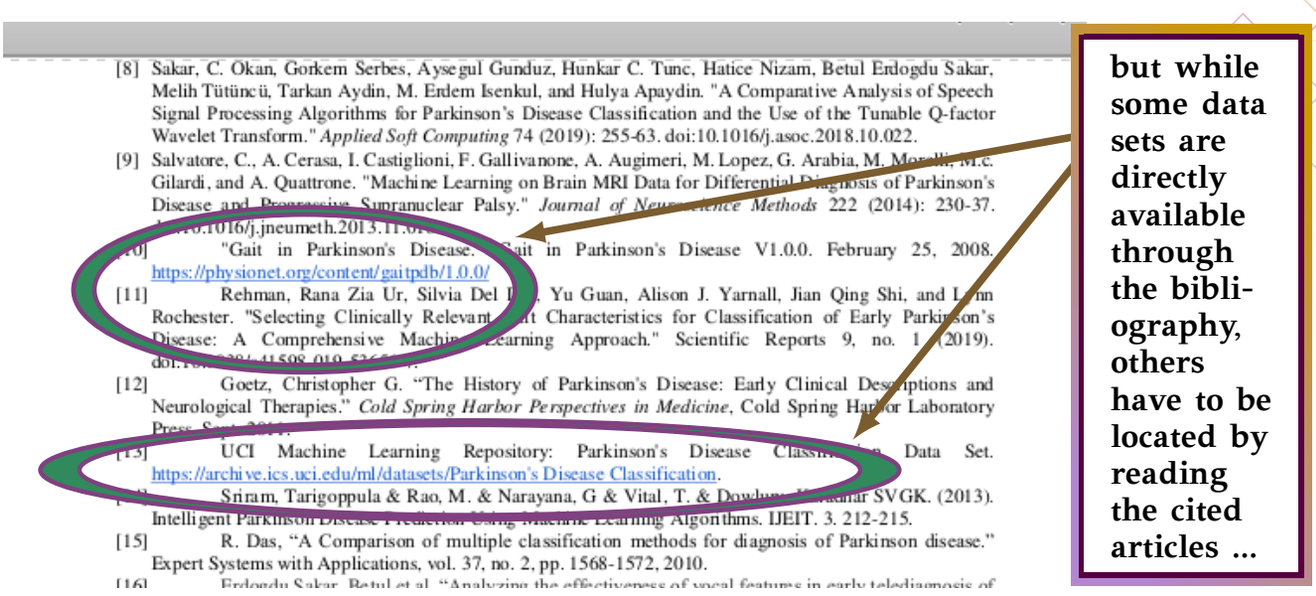

Figure 8: Bibliography (With Data Set Hyperrefs) in the Parkinson's Article

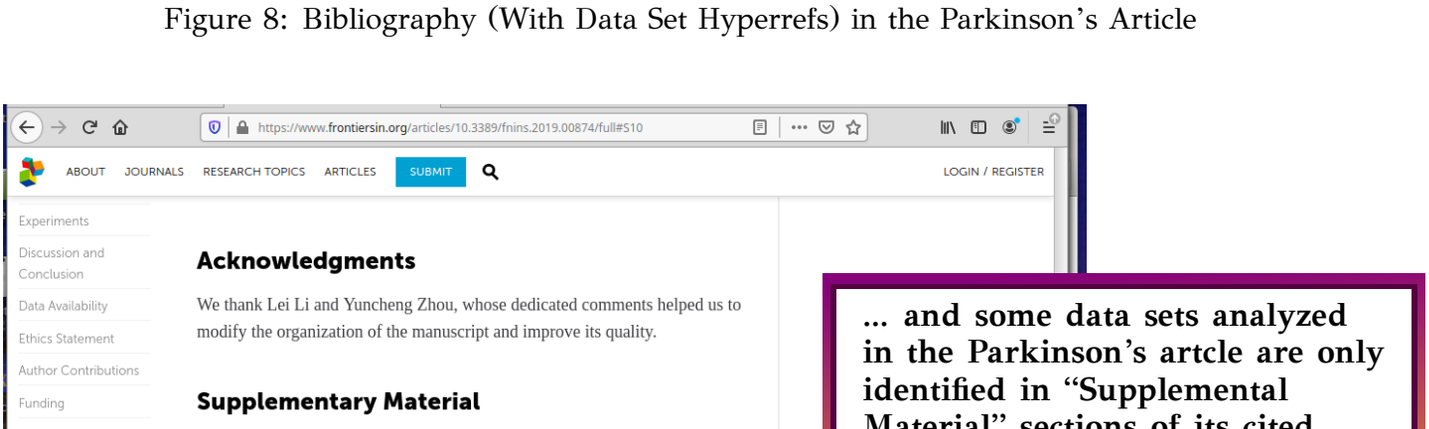

Figure 9: Indirectly Locating Data Sets from Cited Papers

Building Data Sets with ConceptsDB Dataset Creator

In contrast to the paradigm of decentralized curation depicted above, LTS's Dataset Creator can be utilized to build integrated and reusable open-access data sets. With dsC, authors have tools to bundle data with code according to the Research Object protocol and **FAIRSHARING** guidelines, employing an embedded code base to supply a custom desktop-style dataset application that offers a convenient visual, interactive, and analytic interface to the raw data. The **ConceptsDB** database engine, which LTS has designed alongside dsC, can optionally be used to update/manage dsC data. **ConceptsDB** and/or dsC (which may be used together or separately) afford several benefits to publishers, including:

1. Dataset Applications can be configured to link back to the published article, offering an additional route for finding the original article and boosting citations/influence (see Figure 10).
2. Dataset Applications will necessarily be implemented via code libraries encapsulating data access,

libraries which can be leveraged for query engines operating at the data-hosting site, offering more sophisticated and fine-grained query capabilities across multiple data sets. In short, dataset code can be designed to explicate properties of data types, fields, procedures, and file formats, all of which produces queryable metadata that might be leveraged for advanced search features.

3. Documenting dataset applications — via explanations of **GUI** features, compile instructions, test suites, code reviews, and similar semi-formal code representations — can be performed through tools customized for the data-hosting platform, yielding consistent patterns in how data sets are employed, which can benefit human users and also allow some data-acquisition steps to be done automatically.

4. Similarly, dataset applications can expose multiple execution points, thereby supporting workflow description formats. This would allow additional consistency and tooling reuse among multiple data sets sharing a common hosting platform, where workflow representations and implementations could be customized to the platform's architecture and disciplinary foci.

5. Dataset applications can be distributed through platforms such as Qt Creator, Jupyter, SeCo, or ReproZip, yielding publisher-specific tools (e.g., Qt Creator plugins) which can be shared by multiple datasets. This would result in common patterns governing how individual data sets are acquired and accessed, creating a consistency and interoperability among data sets which enhance the value of the overarching hosting platform.

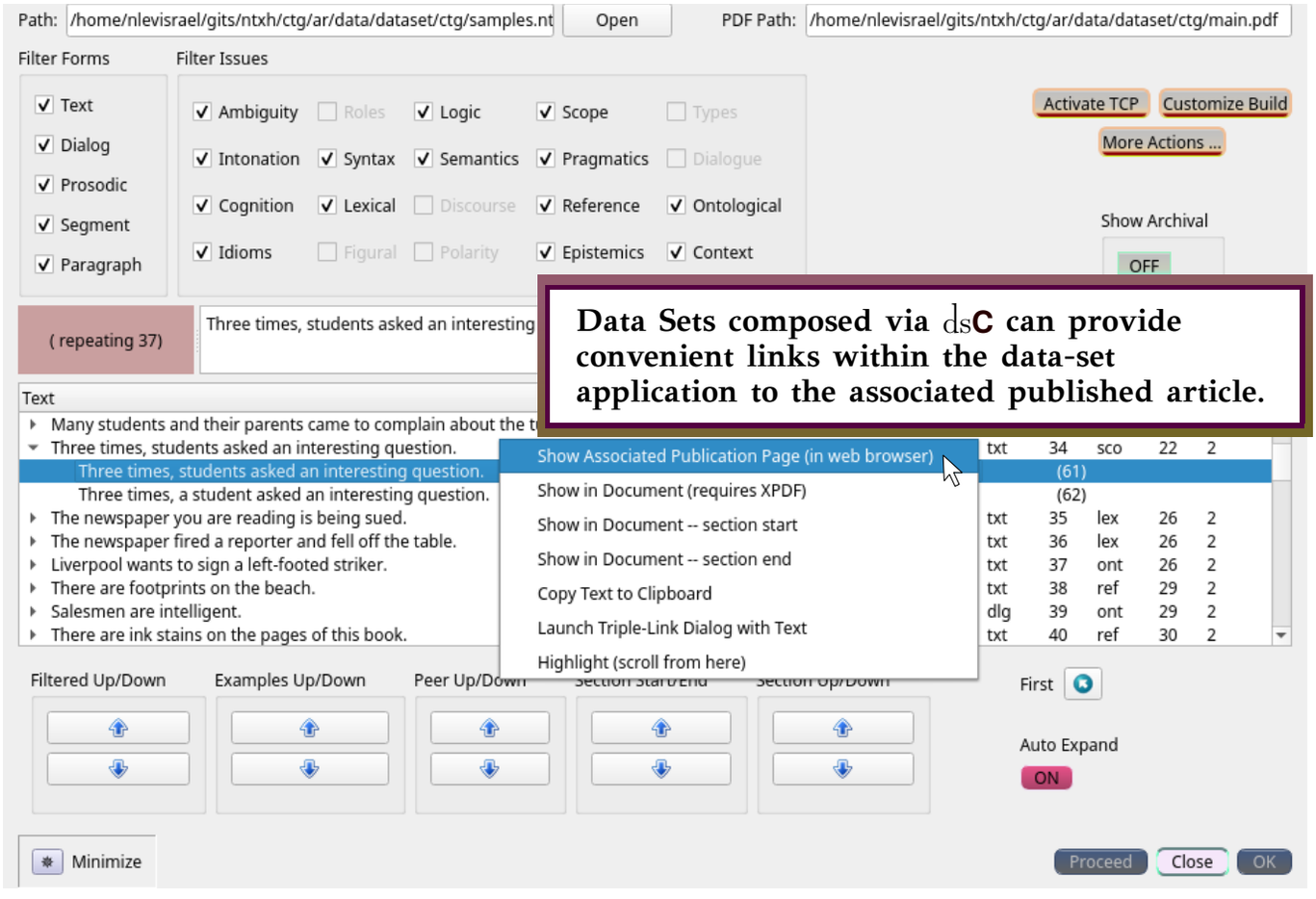

Figure 10: Data Set Linked to Published Article

In sum, data-hosting platforms which provide data sets curated in according with **FAIRSHARING** and Research Object guidelines are more likely to become go-to resources for researchers and educators, because data sets obtained from such platforms are more likely to be interoperable and reusable.

As shown in Figures 10-14, dataset applications also enhance the value of publications, by creating interactive software modules where ideas explored in publication texts are examined or demonstrated using digital assets. Over time, such dataset applications could potentially be reused for different research projects and evolve into general-purpose software tools for the relevant scientific field.

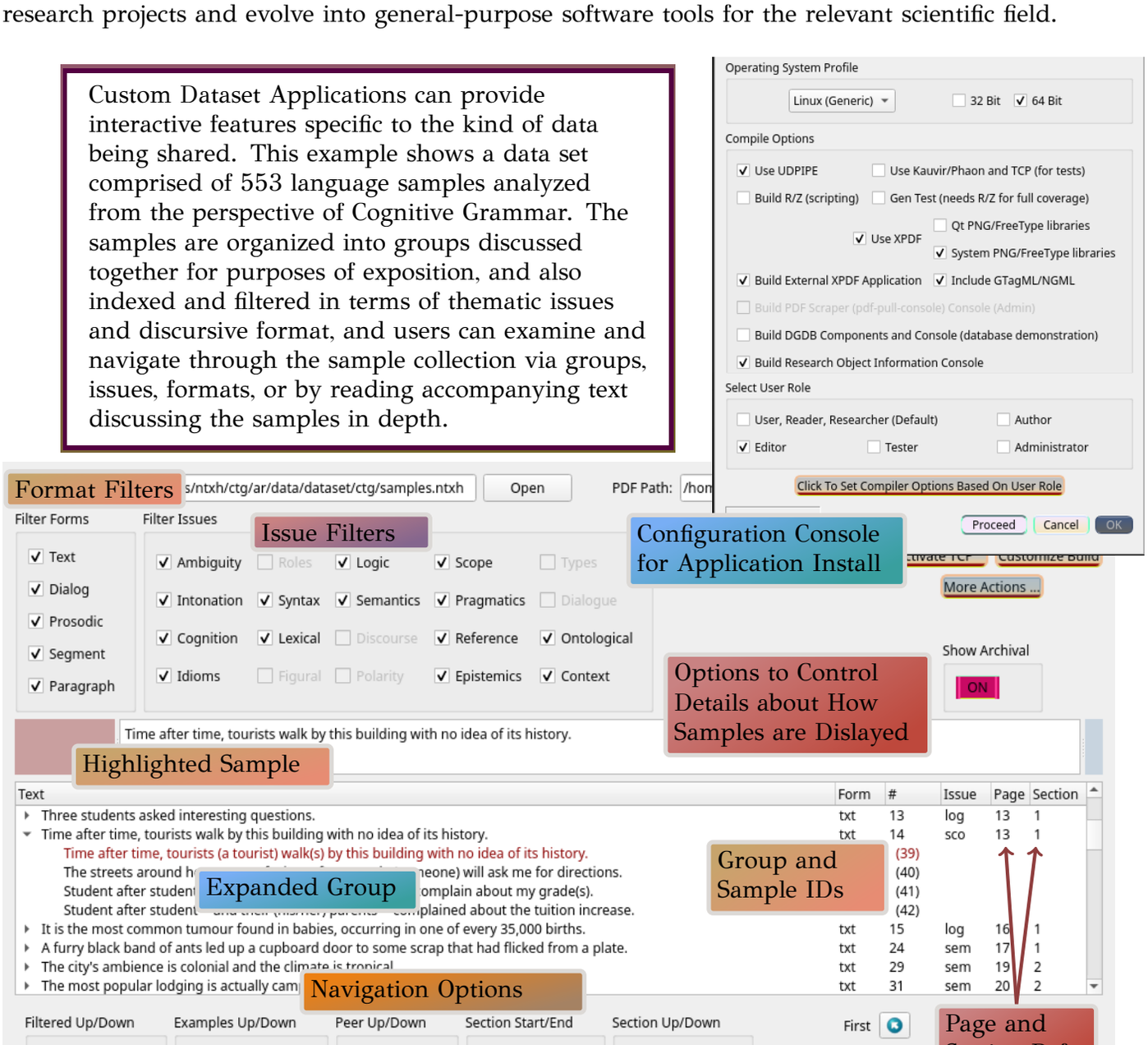

Figure 11: Features of a Custom Dataset Application

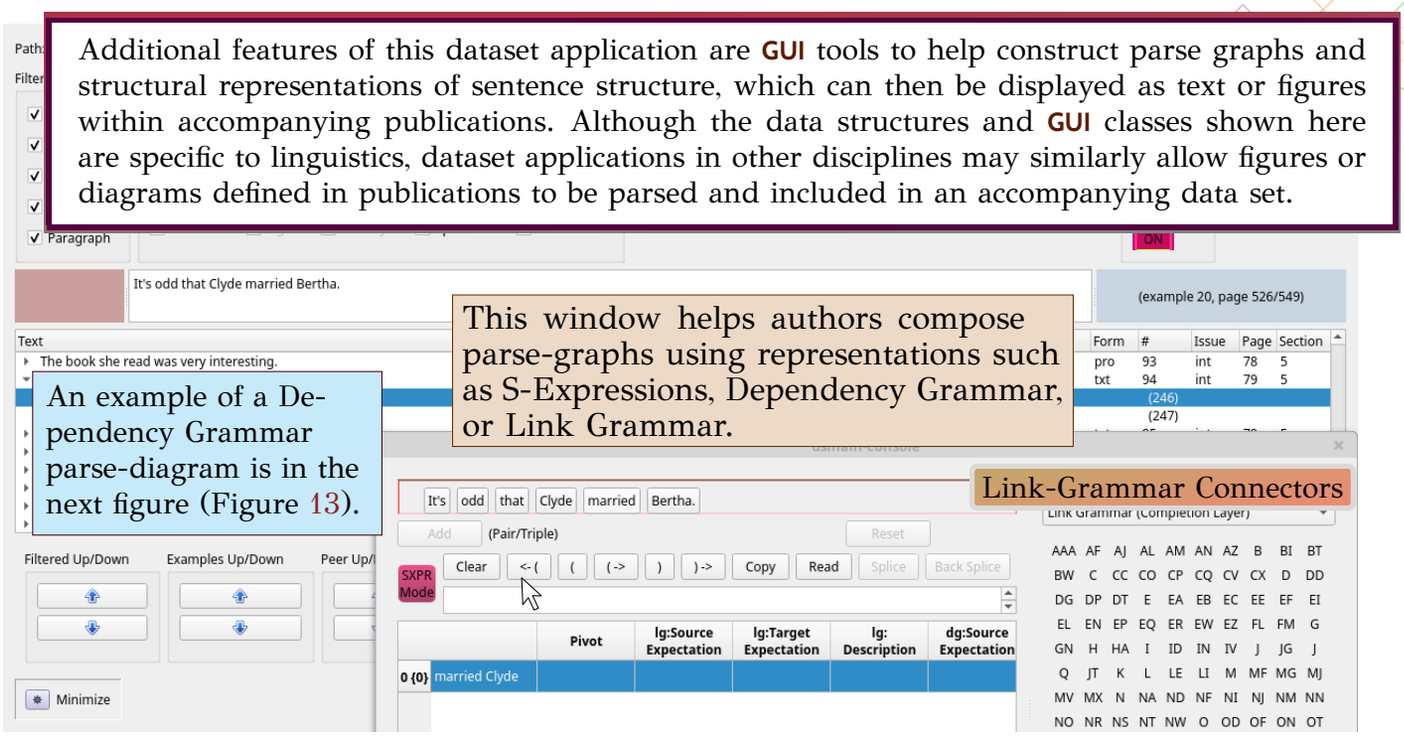

Figure 12: Additional Interactive Features of a Custom Dataset Application

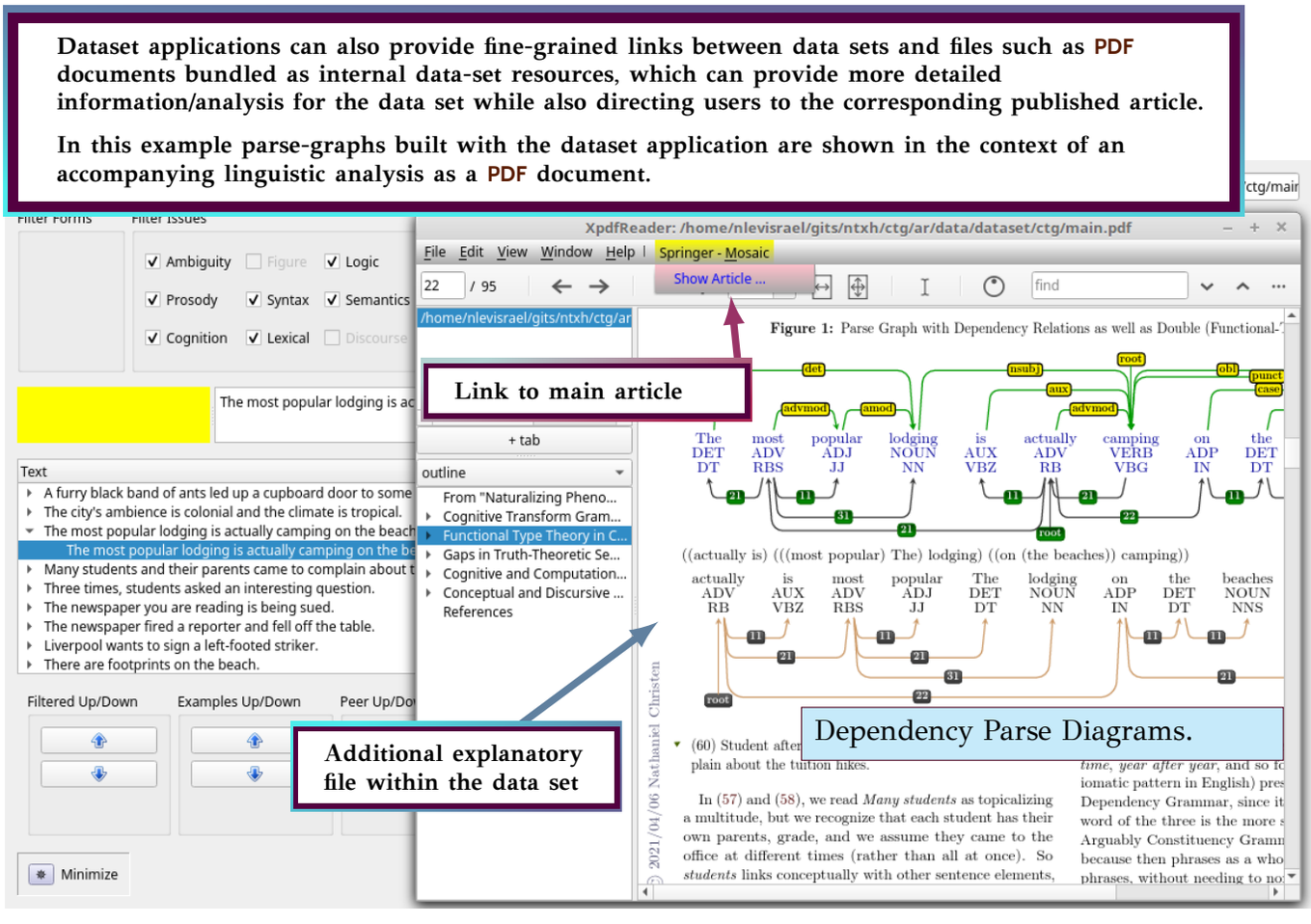

Figure 13: Data Set Linked to PDF File

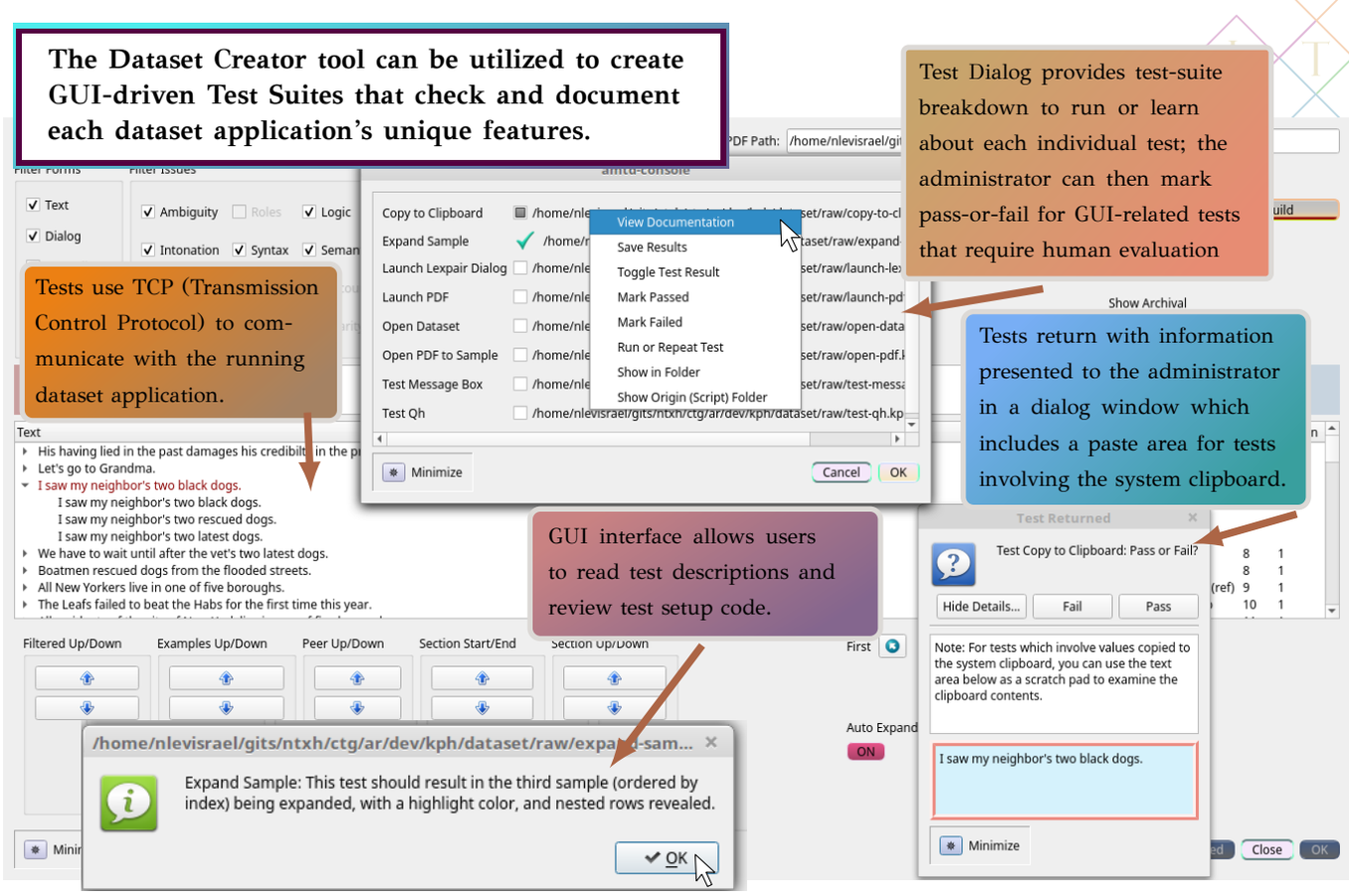

Figure 14: Dataset Application GUI Test Suite

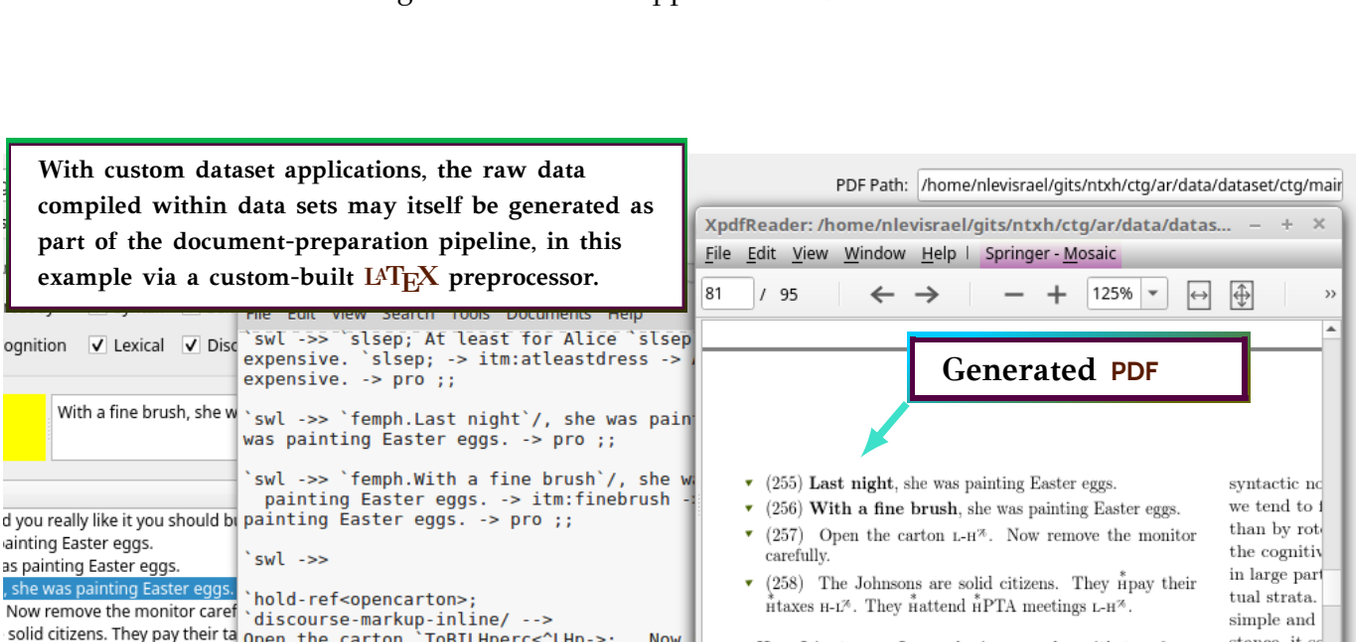

Figure 15: Data Set Extracted from PDF-Generator File

Generating Machine-Readable Publication Text with HTXN

ConceptsDB and dsC employ a novel Hypergraph Text Encoding Protocol (**HTXN**) for text storage/representation. The **HTXN** protocol can also be used independently of the dsC framework for general-purpose document generation. **HTXN** can output **HTML**, LaTeX, and **PDF** files while also producing machine-readable text representations and, concurrently, databases suitable for text mining.

The **HTXN** document-generation pipeline covers multiple stages which aggregate document text and metadata into a queryable resource that recognizes both pure-text content (documents considered as word/sentence/paragraph sequences) and as readable **PDF** files (where sentence and paragraph boundaries are mapped to **PDF** page and screen coordinates). Text and **PDF**-coordinate representations are generated in stages both before and during **PDF** generation, yielding LaTeX auxiliary files which are merged by post-processing algorithms to create an integrated, queryable document model, called a "Semantic Document Infoset" (**SDI**) (see Figure 4). This **SDI** is then grouped into pages and may be loaded by a special **PDF** viewer as each page is shown in the **PDF** window. The **SDI** can then be further annotated to create microcitation links between publications and data sets.

The **SDI** information from many documents could also be deposited in a common database to create an advanced full-text query feature for publication corpora. A detailed overview of **HTXN** and the **SDI** generation algorithms and database persistence options will be provided by LTS on request.

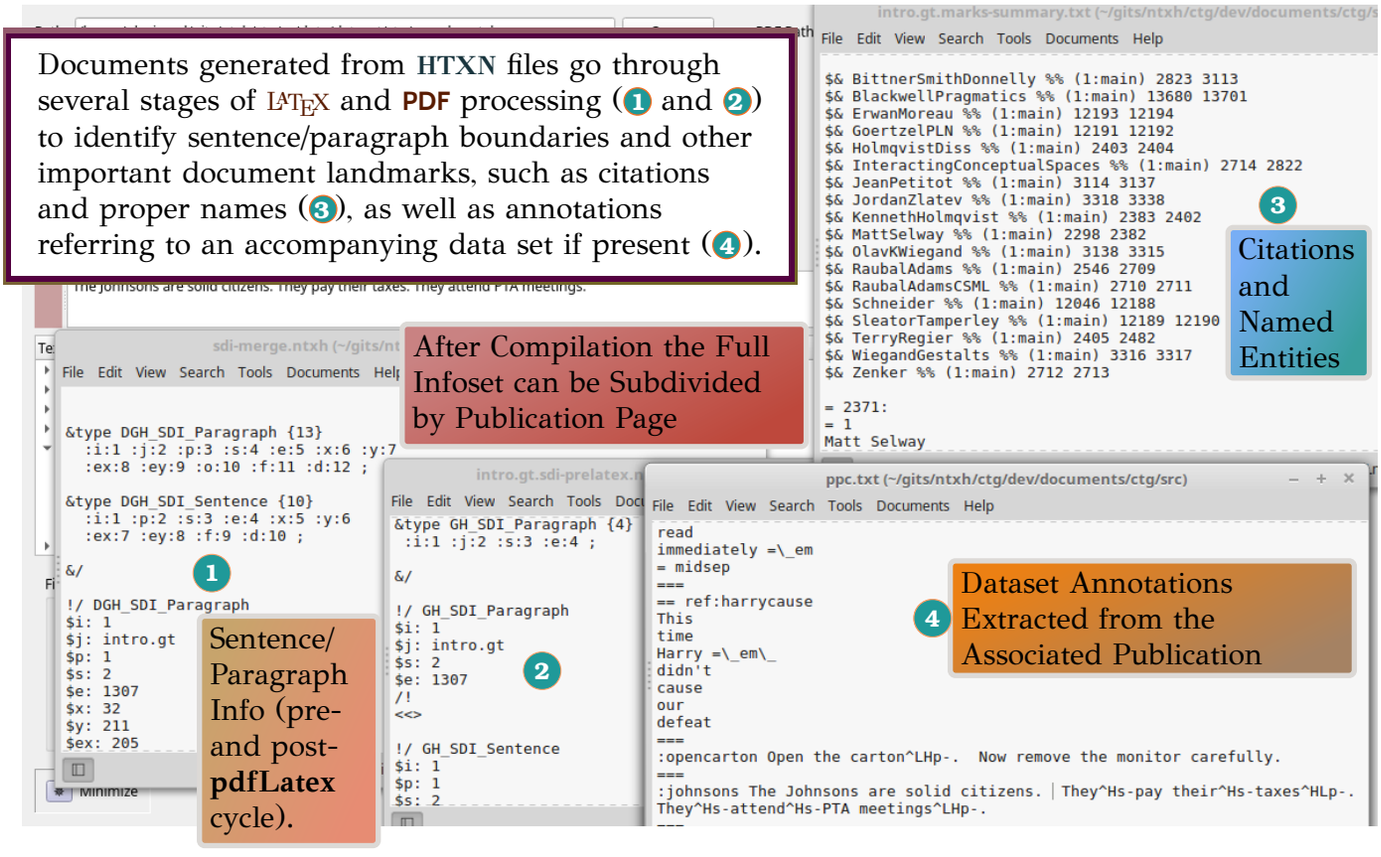

Figure 16: Document-Generation and Machine-Readable Fulltext Derived from HTXN