# Data Gatherer: LLM-Powered Dataset Reference Extraction from Scientific Literature

**Pietro Marini**[1], **Aécio Santos**[1], **Nicole Contaxis**[2], **and Juliana Freire**[1,3]
[1]Tandon School of Engineering
[2]Grossman School of Medicine
[3]Center for Data Science
New York University

## Abstract

Despite growing emphasis on data sharing and the proliferation of open datasets, researchers face significant challenges in discovering relevant datasets for reuse and systematically identifying dataset references within scientific literature. We present Data Gatherer, an automated system that leverages large language models to identify and extract dataset references from scientific publications. To evaluate our approach, we developed and curated two high-quality benchmark datasets specifically designed for dataset identification tasks. Our experimental evaluation demonstrates that Data Gatherer achieves high precision and recall in automated dataset reference extraction, reducing the time and effort required for dataset discovery while improving the systematic identification of data sources in scholarly literature.

## 1 Introduction

The increasing availability of data has accelerated scientific progress. Genomic and proteomic data sharing, for example, has enabled scientists to develop approaches that rely on access to large amounts of data (JB et al., 2020). Policies and frameworks like the FAIR Principles (Wilkinson et al., 2016) and FORCE11's Joint Declaration of Data Citation Principles (Altman et al., 2015) and changing researchers practices have contributed to increasing the amount of data available. Yet, finding datasets for reuse and identifying datasets referenced in research papers remain a challenging and labor-intensive task (Castelo et al., 2021; Borgman and Groth, 2025; Tsueng et al., 2023; Griffiths et al., 2022).

In contrast to journal article and book citation practices that use standardized formats (e.g., citation styles, DOIs), dataset references are inconsistent, ambiguous, and dispersed throughout scholarly documents, making systematic discovery difficult. PubMed and PubMed Central, for example, make some dataset mentions available through LinkOut Resources which links to external resources. They also allow researchers to search for articles that contain associated data in their Data Availability Statement (DAS), a structured section of articles that describe datasets used, or inside similar sections. However, these indexes are not currently able to surface dataset mentions fully, especially those embedded in the article text.

Even when datasets are explicitly referenced, their mentions are often ambiguous. The same dataset may be cited under different names, abbreviations, or project titles across multiple papers. Some papers provide only partial accession codes or omit repository information, making it difficult to resolve the dataset's location. DAS's, for example, may erroneously state that all data from a study is included in the paper (Federer et al., 2018). Common issues like typos, incorrect identifiers, and broken links further hinder discovery.

To locate datasets included in papers, researchers, librarians and data curators then have to undertake the labor-intensive process of manually searching, cross-referencing, and verifying dataset mentions. Mentions may include metadata such as accession codes, repository names, URLs, or informal descriptions. They can be embedded in figure captions, tables, supplementary materials, citations, or structured article sections like a DAS rather than explicitly listed in the main text.

Recent advances in Large Language Models (LLMs) present unprecedented opportunities for automating the discovery and extraction of dataset mentions from scientific literature. LLMs demonstrate superior capability in recognizing complex patterns in natural language text, enabling them to identify dataset references across diverse formats and naming conventions while distinguishing them from superficially similar entities such as gene and experiment identifiers. This can lead to significant improvements in automated extraction.



*Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 114–123

**Contributions.** We introduce Data Gatherer[1], an open-source, LLM-powered system that automates the identification and extraction of structured dataset records from scientific publications. Our system addresses the labor-intensive manual processes currently employed by researchers and librarians for dataset discovery. The design and development of Data Gatherer was informed by a collaboration with biomedical researchers specializing in proteomics and genomics, ensuring the tool addresses real-world requirements. To evaluate the effectiveness of Data Gatherer, we develop two benchmark datasets: (1) a high-quality collection carefully curated and validated by an expert librarian to ensure accuracy and completeness, and (2) a larger-scale dataset constructed through the systematic integration of existing databases that maintain associations between research articles and their referenced datasets. These benchmarks enable a comprehensive evaluation across different scales and quality standards. We present the results of an experimental evaluation which show that Data Gatherer achieves recall of up to 99.4% and precision up to 91.1% across our benchmark datasets.

In summary, our main contributions are: (1) an LLM-powered pipeline for automated identification and extraction of dataset references from scholarly documents (Section 5); (2) development and curation of two benchmark datasets for evaluation of dataset extraction methods (Section 4); and (3) an experimental evaluation of our data extraction methods using different LLMs (Section 6).

## 2 Related Work

The related work on the extraction of dataset mention from scientific literature falls in two main categories, which we describe below.

**Datasets for Information Extraction from Scientific Literature.** Several datasets have been developed to facilitate research in scientific information extraction. Anzaroot and McCallum (2013) introduced a dataset for fine-grained citation field extraction, focusing on segmenting citation strings into components like title and authors. Cheung et al. (2024) presented PolyIE, a dataset for extracting entities and relations specific to polymer materials. Zhang et al. (2024) created SciER, a dataset for entity and relation extraction with a focus on datasets, methods, and tasks. While these datasets facilitate various aspects of scientific information extraction, such as citation parsing and domain-specific entity extraction, we focus on a different task: the extraction of dataset references from the scientific literature on proteomics and genomics.

The Data Citation Corpus, created by Make Data Count in collaboration with the Chan Zuckerberg Initiative, is a comprehensive list of data citations from articles and preprints meant to facilitate the creation and use of data metrics similar to bibliometrics used to measure the impact of other scholarly outputs (e.g., H-index, Impact Factor, and the RCR) (Make Data Count, 2025). The Data Citation Corpus is in part compiled using machine learning methods that leverage SciBERT-based Named Entity Recognition (Istrate, 2023). Make Data Count does not make these data citation location tools publicly available. In contrast, our tool is open source and freely accessible to researchers, and instead of focusing on the creation of data metrics, our goal is to enable users to identify all mentions of datasets within a collection of articles to facilitate data discovery and reuse.

**Dataset Discovery and Citation Analysis.** Early approaches to dataset mention extraction relied on statistical methods. Ghavimi et al. (2016) present a semi-automatic approach combining dictionary-based matching with similarity measures to identify dataset references and link them to existing dataset registries. Zeng and Acuna (2020) propose using a bidirectional LSTM with a CRF inference mechanism for dataset mention detection. Kumar et al. (2021) propose DataQuest, a BERT-based entity recognition model with POS-aware embeddings, utilizing a two-stage pipeline for dataset sentence classification and mention extraction. These methods face important limitations that constrain their practical applicability. First, they typically require domain-specific training data, limiting their transferability across research disciplines. Second, the relatively small model sizes and training corpora restrict their ability to capture the full diversity of dataset naming conventions and referencing patterns found in scientific literature. Third, these methods often struggle with implicit or contextual dataset references that require deeper semantic understanding beyond surface-level pattern matching. Our work addresses these limitations by leveraging the robust information extraction capabilities of large language models trained on extensive, diverse corpora. This approach enables more generalizable extraction across domains while reducing dependence on manually curated training data.

[1] https://github.com/VIDA-NYU/data-gatherer

## 3 Problem Definition

We aim to automatically discover and extract dataset references from scholarly publications, focusing on citations accessible in academic documents available on the Web.

**Definition 1.** Given a publication $P$ (e.g., a URL or DOI that refers to a scholarly article), the goal is to build a function $\mathcal{F}$ that extracts a structured set of records $\{(d_i, r_i)\}$ from $P$, i.e., $\mathcal{F}(P) = \{(d_1, r_1), (d_2, r_2), ..., (d_n, r_n)\}$, where $d_i$ is the dataset identifier, typically an accession code or another type of dataset reference, and $r_i$ is the repository name or reference (e.g., a plain text string or a URL pointing to the repository). □

We consider a dataset reference valid if its identifier $d_i$ exists in the repository $r_i$. To evaluate the ability of different approaches to identify and extract valid dataset references correctly, we built two benchmark datasets that are detailed in Section 4.

## 4 The DataRef Benchmarks

To evaluate Data Gatherer, we constructed two datasets using distinct methodologies: (1) DataRef-EXP was manually curated by an expert librarian who identified and reviewed publication web pages on PubMed Central, selecting articles to ensure a diverse representation of dataset citation formats; (2) DataRef-REV was built by combining metadata from two online resources: ProteomeCentral,[2] a portal that aggregates dataset information from repositories within the ProteomeXchange consortium (Deutsch et al., 2023), and the Gene Expression Omnibus (GEO) repository.[3] Below we detail the data curation approach for each of these datasets. The datasets are available for download at https://doi.org/10.5281/zenodo.15549086.

### 4.1 DataRef-EXP Dataset

The DataRef-EXP dataset was created through systematic manual selection and curation of scholarly journal articles to ensure a comprehensive representation of dataset citation formats and referencing patterns. The articles were exclusively sourced from PubMed Central (PMC)[4] for two important reasons. First, PMC provides open access to the full text of articles via an API, eliminating potential copyright restrictions and technical barriers to systematically download journal articles. Second, PMC's advanced filtering capabilities enable targeted identification of articles containing explicit data references, particularly through their Data Availability Statements (DAS). DAS explicitly document the datasets employed in research and provides access information or retrieval instructions. While DAS quality varies across publications—with many exhibiting incomplete metadata or outdated access links—their presence serves as a reliable indicator for articles likely to contain dataset references. This filtering mechanism streamlined the curation process by pre-selecting articles with higher probability of containing relevant dataset mentions.

A total of 21 journal articles were selected, containing 48 dataset references. Journal articles were chosen in order to maximize the variation in how included datasets were referenced, enabling a comprehensive evaluation of the Data Gatherer's ability to extract dataset mentions across various formats. For example, some journal articles were chosen in which all dataset mentions were included in the DAS while other journal articles included dataset mentions in figures, in tables, or within the text. Additionally, some articles were chosen due to errors in dataset mentions, like inaccurate accession numbers or incomplete dataset information (e.g., an accession number but no named repository).

### 4.2 DataRef-REV Dataset

The second benchmark dataset was constructed using a systematic reverse-engineering methodology that leverages structured metadata exports from established scientific data repositories: ProteomeCentral and Gene Expression Omnibus (GEO). ProteomeCentral is a valuable source for ground truth data, offering curated metadata for 23,348 publicly available datasets, including dataset identifiers and one or more valid related paper DOIs or PubMed identifiers. It aggregates datasets from various repositories and links them to citing publications, making it a great starting point for locating papers that contain dataset references. Similarly, GEO is a public functional genomics data repository managed by the National Center for Biotechnology Information (NCBI). GEO provides programmatic access through a REST API that allowed us to retrieve 165,078 dataset identifiers with valid references to publications that mention them.

A limitation of DataRef-REV is that it only contains references to datasets from repositories that are part of the ProteomeXchange consortium or

[2] https://proteomecentral.proteomexchange.org/
[3] https://www.ncbi.nlm.nih.gov/geo/
[4] https://pmc.ncbi.nlm.nih.gov/

the GEO – it is possible that there may be datasets mentioned in the paper that are not deposited in these repositories. However, an advantage is that it is automatically generated, allowing us to obtain a much larger number of dataset references compared to `DataRef-EXP` (manually curated).

**Dataset Construction Details.** Each dataset entry includes a unique identifier, typically an accession code, along with the corresponding repository name, such as PRIDE, MassIVE, iProX, jPOST, PeptideAtlas, or PanoramaPublic. Additionally, the metadata contains information about citing publications, including their DOI or PubMed Central ID (PMCID) when available, as well as the title and keywords associated with the dataset. To ensure high-quality metadata, entries lacking a DOI or publication link were discarded, guaranteeing that each dataset-reference pair has an associated paper reference. As a result, `DataRef-REV` contains 397,263 dataset references records from 244,847 papers to 188,426 datasets.

To supplement the structured metadata, we implemented an automated data-fetching pipeline to retrieve full-text HTML versions of citing publications. Using Selenium, we systematically accessed publisher web sites and extracted the HTML source of each article when it was available. By integrating full-text data with structured repository metadata, we ensure that our dataset reflects both formally registered dataset citations and real-world citation practices in scholarly writing.

## 5 The Data Gatherer Tool

Data Gatherer was designed to automatically extract dataset references from scientific publications by processing both HTML web pages and XML responses as discussed in Section 3. It employs LLMs to identify references and construct links that enable the retrieval of the dataset. We have explored two main strategies: Full-Document Read (FDR) and Retrieve-Then-Read (RTR).

### 5.1 Retrieve-Then-Read (RTR)

The RTR method is a two-step process designed to improve efficiency in dataset reference extraction by leveraging the structural elements of full-text documents for scientific articles. It first locates specific target sections of the papers where dataset mentions are likely to appear, such as the DAS and similar sections. Then, it collects the textual content from the target sections and feeds them to an LLM using a few-shot prompt to extract dataset references (we provide prompts in the Appendix 7). We use the RTR approach for two main reasons. First, by drastically reducing the input length, RTR lowers both inference time and computational cost. Second, if the retrieval step is effective, it preserves most of the relevant information needed for extraction, allowing the language model to focus on likely regions of interest. However, retrieval must be precise: naïve or hard-coded retrieval rules may miss the critical passages and lead to lower recall.

Since RTR relies on structured documents, it currently applies only to open-access articles from PubMed Central (PMC), but the method can be extended to other sources with similar structural cues.

**Rule-Based Section Retrieval.** We developed a rule-based retrieval method to identify the sections of the raw XML/HTML documents that are likely to contain references to the dataset. It uses a combination of CSS selectors and XPath expressions, which we defined after various trial-and-error experiments on publications comprising diverse forms of dataset citation records. The retrieval rules, which are configured in a JSON file, are organized in two levels: general rules apply to the raw input data regardless of the specific publisher, and the remaining rules are tailored for use only in specific domains.

**LLM-Based Dataset Extraction.** Following section retrieval, we apply LLM-based extraction and instruct the LLM to output dataset references in JSON format using a structured few-shot prompting approach. Multiple prompt variations were tested and refined to improve extraction precision.

### 5.2 Full-Document Read (FDR)

To avoid the costs associated with manually defining rules for locating target sections, we consider an alternative approach that utilizes an LLM-based extraction pipeline to process the entire document. Instead of processing only specific sections of the article, we use the entire document text. While this method is more adaptable to various publishers, it has some drawbacks. Specifically, it only works with LLMs that support relatively long context windows and requires them to handle a significantly larger input, which increases costs.

**HTML Preprocessing & Filtering.** Before passing documents to the LLM, we perform an HTML normalization step to remove non-informative elements, such as scripts, styles, images, iframes, buttons, and metadata tags. This preprocessing ensures that only relevant text-based content is considered, reducing noise and improving dataset ex-

traction performance and costs.

**Handling Long HTML Documents.** We use only LLMs that support long-context windows, with `gpt-4o-mini` (128K tokens) being the model with the smallest context limit. In cases where documents exceed this limit, the content is truncated until it fits the model context size constraints.

## 6 Experimental Evaluation

We evaluated Data Gatherer's performance using `DataRef-EXP` and `DataRef-REV` (Section 4). Due to the size of the `DataRef-REV` dataset, we used a sample consisting of 1,883 dataset citation records from 1,242 PubMed Central articles. Performance was assessed by using precision and recall metrics calculated for each paper and then averaged over the set of papers to compute the average precision and average recall. Since identifiers are typically unique across repositories (e.g., DOIs), we compare only the identifiers to determine matches. To account for common identifier variations (e.g., DOI: `10.6019/PXD123456` vs. Accession Code: `PXD123456`), we consider both exact or partial matches (e.g., substring match) for dataset identifiers.

We report the results in Table 1, which includes a comparison of different LLMs and extraction methods for the two datasets. Both methods (FDR and RTR) attain high precision and recall. `gpt-4o-mini` attains higher precision than `gemini-2.0-flash` for both methods on the `DataRef-EXP` dataset, but not for `DataRef-REV`. Note that the *maximum* recall on `DataRef-EXP` is generally lower, which is expected since the dataset was designed to include a high variety of difficult cases. Moreover, the RTR method struggles to obtain high recall in `DataRef-REV` dataset, potentially due to the low coverage of the manually curated rules, which may lead to missing important parts of the input. Despite of the low recall, the RTR method seems to improve precision in some cases.

While not conclusive, these results suggest that reducing the input size can help improve the cost (due to smaller input size) and the precision of long-context models (at the cost of decreasing the recall in some cases). Thus, more accurate and general RTR methods could be beneficial to improve the overall results. We also note that the results reported on the `DataRef-REV` dataset are limited, specially precision, since it may be possible that the models identify correct datasets that are not included in the ground truth (see section 4).

| Dataset | Model | Method | Precision | Recall |
|---|---|---|---|---|
| `DataRef-EXP` | gpt-4o-mini | FDR | 0.843 | 0.821 |
| | | RTR | **0.911** | **0.905** |
| | gemini-2.0-flash | FDR | 0.704 | **0.817** |
| | | RTR | **0.880** | 0.802 |
| `DataRef-REV` | gpt-4o-mini | FDR | **0.853** | **0.985** |
| | | RTR | 0.684 | 0.635 |
| | gemini-2.0-flash | FDR | 0.754 | **0.994** |
| | | RTR | **0.803** | 0.563 |

Table 1: Comparison of different LLMs, and methods (FDR, RTR) on `DataRef-EXP` vs `DataRef-REV`.

## 7 Conclusion

Researchers, librarians, and data curators currently spend significant amounts of time locating dataset mentions in scholarly papers. They perform this work both to locate datasets for secondary analysis projects and also to ensure a paper's conclusions are well-supported by the data. To ease this time-intensive and difficult task, we designed Data Gatherer to automatically find and parse dataset mentions in articles. As new methodologies in the sciences increasingly rely on access to large amounts of open data this tool can have a notable impact on the way that researchers, data curators, and librarians find, review, and aggregate data to meet the promise of these new methods.

## Limitations

Our work has several limitations. The retrieve-then-read (RTR) approach only supports the PubMed Central (PMC) structure, so it requires extra effort to extend it to other repositories. The full-document read (FDR) approach aims to resolve this limitation by processing the full document, however, this limits the number of LLMs that can be used and may increase processing costs. Regardless of the strategy, the system can miss dataset references or output incorrect references. It also relies on LLM capabilities, which can be limited in ambiguous contexts. Additionally, our evaluation datasets, `DataRef-EXP` and `DataRef-REV`, may not fully represent all dataset citation practices since their size is limited and mainly cover papers related to proteomics and genomics research fields.

## Acknowledgements

This work was supported by NSF awards IIS-2106888 and OAC-2411221, the DARPA ASKEM program Agreement No. HR0011262087, and the ARPA-H BDF program. The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the DARPA, ARPA-H, the U.S. Government, or NSF.

## References


Micah Altman, Christine Borgman, Mercè Crosas, and Maryann Matone. 2015. An introduction to the joint principles for data citation. *Bulletin of the Association for Information Science and Technology*, 41(3):43–45.

Sam Anzaroot and Andrew McCallum. 2013. A new dataset for fine-grained citation field extraction. In *ICML Workshop on Peer Reviewing and Publishing Models (PEER)*.

Christine L. Borgman and Paul Groth. 2025. From Data Creator to Data Reuser: Distance Matters. *Harvard Data Science Review*, 7(2). Https://hdsr.mitpress.mit.edu/pub/2mvqwgmf.

Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment*, 14(12):2791–2794.

Jerry Cheung, Yuchen Zhuang, Yinghao Li, Pranav Shetty, Wantian Zhao, Sanjeev Grampurohit, Rampi Ramprasad, and Chao Zhang. 2024. PolyIE: A dataset of information extraction from polymer material scientific literature. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2370–2385.

Eric W Deutsch, Nuno Bandeira, Yasset Perez-Riverol, Vagisha Sharma, Jeremy J Carver, Luis Mendoza, Deepti J Kundu, Shengbo Wang, Chakradhar Bandla, Selvakumar Kamatchinathan, and 1 others. 2023. The proteomexchange consortium at 10 years: 2023 update. *Nucleic Acids Research*. PMID: 36370099, DOI: https://doi.org/10.1093/nar/gkac1040.

Lisa M. Federer, Christopher W. Belter, Douglas J. Joubert, Alicia Livinski, Ya-Ling Lu, Lissa N. Snyders, and Holly Thompson. 2018. Data sharing in PLOS ONE: An analysis of data availability statements. *PLOS ONE*, 13(5):e0194768.

Behnam Ghavimi, Philipp Mayr, Sahar Vahdati, and Christoph Lange. 2016. Identifying and improving dataset references in social sciences full texts. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 105–114. IOS Press.

Emily Griffiths, Rebecca M Joseph, George Tilston, Sarah Thew, Zoher Kapacee, William Dixon, and Niels Peek. 2022. Findability of UK health datasets available for research: a mixed methods study. *BMJ Health Care Inform*, 29(1):e100325.

Ana-Maria Istrate. 2023. Building the Open Global Data Citation Corpus – Chan Zuckerberg Initiative. Publisher: Zenodo Version Number: 1.0.

Byrd JB, Greene AC, Prasad DV, Jiang X, and Greene CS. 2020. Responsible, practical, genomic data sharing that accelerates research. *Nature Reviews Genetics*.

Sandeep Kumar, Tirthankar Ghosal, and Asif Ekbal. 2021. DataQuest: An approach to automatically extract dataset mentions from scientific papers. In *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1–3, 2021, Proceedings 23*, pages 43–53. Springer.

Make Data Count. 2025. Open data metrics require open infrastructure: Data Citation Corpus. https://makedatacount.org/find-a-tool/.

Ginger Tsueng, Marco A. Alvarado Cano, José Bento, Candice Czech, Mengjia Kang, Lars Pache, Luke V. Rasmussen, Tor C. Savidge, Justin Starren, Qinglong Wu, Jiwen Xin, Michael R. Yeaman, Xinghua Zhou, Andrew I. Su, Chunlei Wu, Liliana Brown, Reed S. Shabman, Laura D. Hughes, the NIAID Systems Biology Data Dissemination Working Group, and Serdar Turkarslan. 2023. Developing a standardized but extendable framework to increase the findability of infectious disease datasets. *Scientific Data*, 10(1):99.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino Da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, and 34 others. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018.

Tong Zeng and Daniel Acuna. 2020. Finding datasets in publications: the syracuse university approach. In *Rich Search and Discovery for Research Datasets*, pages 158–165. SAGE.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. 2024. SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13083–13100.


## Appendix: LLM Prompts

The prompts used in our experiments for both the Retrieve-Then-Read (RTR) method are given in Figure 1 and Figure 2, while Figure 3 and Figure 4 show the prompts used in the FDR method.

**Role:** system
**Content:**
You are a specialized assistant that extracts dataset references from the content of scientific papers. You must output a JSON array of objects, where each object has the following keys: 'dataset_identifier', 'data_repository', and 'dataset_webpage'. Follow the structure of the provided examples exactly.

---

**Role:** user
**Content:**
Extract dataset references based on the examples below:

Example 1:
Content: "The study used dataset EGAS00001000925, which is available at the European Genome Archive."
Response:

```
[
  {
    "dataset_identifier": "EGAS00001000925",
    "data_repository": "European Genome Archive",
    "dataset_webpage": "https://ega-archive.org/studies/EGAS00001000925"
  }
]
```

Example 2:
Content: "Proteomics data was obtained from PRIDE, accession PXD029821."
Response:

```
[
  {
    "dataset_identifier": "PXD029821",
    "data_repository": "PRIDE",
    "dataset_webpage": "https://www.ebi.ac.uk/pride/archive/projects/PXD029821"
  }
]
```

Example 3:
Content: "The repository dbGaP hosts the dataset phs001366.v1.p1 at this location."
Response:

```
[
  {
    "dataset_identifier": "phs001366.v1.p1",
    "data_repository": "dbGaP",
    "dataset_webpage": "https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.
    cgi?study_id=phs001366.v1.p1"
  }
]
```

Now process the following content:

Content: {content}

Figure 1: Prompt for Gemini to extract dataset references from small HTML elements, used for the RTR method.

**Role:** system
**Content:**
You are a specialized assistant that extracts dataset references from the content of scientific papers. You must output a JSON array of objects, where each object has the following keys: 'dataset_identifier', 'data_repository', and 'dataset_webpage'. Follow the structure of the provided examples exactly. We will not wrap the json codes in JSON markers.

---

**Role:** user
**Content:**
Extract dataset references based on the examples below:

Example 1:
Content: "The study used dataset EGAS00001000925, which is available at the European Genome Archive."
Response:

```
[
  {
    "dataset_identifier": "EGAS00001000925",
    "data_repository": "European Genome Archive",
    "dataset_webpage": "https://ega-archive.org/studies/EGAS00001000925"
  }
]
```

Example 2:
Content: "Proteomics data was obtained from PRIDE, accession PXD029821."
Response:

```
[
  {
    "dataset_identifier": "PXD029821",
    "data_repository": "PRIDE",
    "dataset_webpage": "https://www.ebi.ac.uk/pride/archive/projects/PXD029821"
  }
]
```

Example 3:
Content: "The repository dbGaP hosts the dataset phs001366.v1.p1 at this location."
Response:

```
[
  {
    "dataset_identifier": "phs001366.v1.p1",
    "data_repository": "dbGaP",
    "dataset_webpage": "https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.
    cgi?study_id=phs001366.v1.p1"
  }
]
```

Now process the following content:

Content: {content}

Figure 2: Prompt for GPT to extract dataset references from small HTML elements, used for the RTR method.

**Role:** model
**Content:**
I am a large language model trained to be informative and comprehensive. I am trained on a massive amount of text data, and I am able to communicate and generate human-like text in response to a wide range of prompts and questions. For this task, I will act as a specialized assistant that can identify datasets mentioned in a publication and create a summary suitable for non-specialists.
The output should be a JSON array of objects, where each object has the following keys:
- "dataset_identifier": This is any alphanumeric string (maybe including punctuation marks) that uniquely identifies or provides access to a dataset.
- "repository_reference": This is the URL or reference to the data repository where the dataset can be found.
Here are some examples for reference:

```
[
  'dataset_identifier' => 'EGAS00001000925',
  'repository_reference' => 'https://ega-archive.org/datasets/EGAS00001000925',

  'dataset_identifier' => 'GSE69091',
  'repository_reference' => 'https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=
    GSE69091',

  'dataset_identifier' => 'PRJNA306801',
  'repository_reference' => 'https://www.ncbi.nlm.nih.gov/bioproject/?term=
    PRJNA306801',

  'dataset_identifier' => 'phs003416.v1.p1',
  'repository_reference' => 'dbGaP',

  'dataset_identifier' => 'PXD049309',
  'repository_reference' => 'https://www.ebi.ac.uk/pride/archive/projects/
    PXD049309',

  'dataset_identifier' => 'IPX0004230000',
  'repository_reference' => 'http://www.iprox.org',

  'dataset_identifier' => 'MSV000092944',
  'repository_reference' => 'https://massive.ucsd.edu/',

  'dataset_identifier' => 'n/a',
  'repository_reference' => 'https://data.broadinstitute.org/ccle_legacy_data/
    mRNA_expression/'
]
```

---

**Role:** user
**Content:** Given the information that I am going to share:
1) the webpage in HTML format that you have to extract datasets information from.
2) a sample of already known data repositories.
Please return a JSON array of objects where each object has the following structure:
- 'dataset_identifier': The dataset identifier (a code). If not found, set it to "n/a".
- 'repository_reference': The URL or reference to the data repository. If not found, set it to "n/a".
Please follow these strict instructions:
- The output must be a valid JSON array of objects.
- Each object must contain the keys 'dataset_identifier' and 'repository_reference'.
- Any other output format will be considered invalid.
Below is the input data that you will use to generate the output:

1) html => {content}

2) repos => {repos}

Figure 3: Prompt for Gemini to extract dataset references from full documents normalized, used for the FDR method.

**Role:** system
**Content:**
I am a large language model trained to be informative and comprehensive. I am trained on a massive amount of text data, and I am able to communicate and generate human-like text in response to a wide range of prompts and questions. For this task, I will act as a specialized assistant that can identify datasets mentioned in a publication and create a summary suitable for non-specialists.
The output should be a JSON array of objects, where each object has the following keys:
- ‘dataset_identifier‘: This is any alphanumeric string (maybe including punctuation marks) that uniquely identifies or provides access to a dataset.
- ‘repository_reference‘: This is the URL or reference to the data repository where the dataset can be found.
Here are some examples for reference:

```
[
  'dataset_identifier' => 'EGAS00001000925',
  'repository_reference' => 'https=>//ega-archive.org/datasets/EGAS00001000925',

  'dataset_identifier' => 'GSE69091',
  'repository_reference' => 'Gene Expression Omnibus (GEO)',

  'dataset_identifier' => 'PRJNA306801',
  'repository_reference' => 'https=>//www.ncbi.nlm.nih.gov/bioproject/?term=
    PRJNA306801',

  'dataset_identifier' => 'phs003416.v1.p1',
  'repository_reference' => 'dbGaP',

  'dataset_identifier' => 'PXD049309',
  'repository_reference' => 'https=>//www.ebi.ac.uk/pride/archive/projects/
    PXD049309',

  'dataset_identifier' => 'IPX0004230000',
  'repository_reference' => 'http=>//www.iprox.org',

  'dataset_identifier' => 'MSV000092944',
  'repository_reference' => 'https=>//massive.ucsd.edu/',

  'dataset_identifier' => 'n/a',
  'repository_reference' => 'https://data.broadinstitute.org/ccle_legacy_data/
    mRNA_expression/'
]
```

---

**Role:** user
**Content:**

I have a webpage in HTML format ({content}) and a list of known data repositories ({repos}). Please return a JSON array of objects, where each object has the structure:
- ‘dataset_id’: The dataset identifier (a code). If not found, set it to ‘n/a’.
- ‘repository_reference’: The URL or reference to the data repository. If not found, set it to ‘n/a’.

Ensure the output is a plain JSON array, not nested inside another structure, and not an Unterminated string.

Input:
content => {content}
repos => {repos}

Figure 4: Prompt for GPT to extract dataset references from full documents normalized, used for the FDR method.