DataDoc Analyzer: A Tool for Analyzing the Documentation of Scientific Datasets

Joan Giner-Miguelez
Internet Interdisciplinary Institute, IN3
Universitat Oberta de Catalunya, UOC
Barcelona, Spain
jginermi@uoc.edu

Abel Gómez
Internet Interdisciplinary Institute, IN3
Universitat Oberta de Catalunya, UOC
Barcelona, Spain
agomezlla@uoc.edu

Jordi Cabot
Luxembourg Institute of Science and Technology (LIST)
Esch-Sur-Alzette, Luxembourg
jordi.cabot@list.lu

ABSTRACT

Recent public regulatory initiatives and relevant voices in the ML community have identified the need to document datasets according to several dimensions to ensure the fairness and trustworthiness of machine learning systems. In this sense, the data-sharing practices in the scientific field have been quickly evolving in the last years, with more and more research works publishing technical documentation together with the data for replicability purposes. However, this documentation is written in natural language, and its structure, content focus, and composition vary, making them challenging to analyze.

We present DataDoc Analyzer, a tool for analyzing the documentation of scientific datasets by extracting the details of the main dimensions required to analyze the fairness and potential biases. We believe that our tool could help improve the quality of scientific datasets, aid dataset curators during its documentation process, and be a helpful tool for empirical studies on the overall quality of the datasets used in the ML field. The tool implements an ML pipeline that uses Large Language Models at its core for information retrieval. DataDoc is open-source, and a public demo is published online.

CCS CONCEPTS

• **Artificial Intelligence** → *Data Science*; Fairness; • **Software Engineering** → Data Quality.

KEYWORDS

Datasets, Machine learning, Fairness , Reverse Engineering, Large Language Models, Explainability

ACM Reference Format:
Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2023. DataDoc Analyzer: A Tool for Analyzing the Documentation of Scientific Datasets. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3614737

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614737

1 INTRODUCTION

The increasing impact of machine learning technologies on our society has raised the interest of researchers and regulatory agencies in the fairness and trustworthiness of these systems. Datasets play a central role in those systems, and recent works in the ML community have identified the need to document the data on several dimensions, such as the context where the data has been collected or annotated, or the social impact on specific groups [4, 9, 1, 5]. For instance, medical datasets imbalanced in terms of gender could produce biased classifiers for computer-aided diagnosis [7], or language datasets gathered from Australian speakers could drop the accuracy of models trained to support users in the United States because of the different language styles [1].

On the other hand, recent public regulatory initiatives such as the European AI Act[1] and the AI Right of Bills[2] also recognize the need to provide technical documentation about the data used to train ML models and the context in which these data have been curated. These same agencies also call for this documentation to be easy to understand by non-experts to bridge the gap between technology and end users.

In that sense, data-sharing practices in the scientific field have been quickly evolving in the last few years [11]. The adoption of Data Management Plans [3] by research institutions and the creation of scientific data journals have motivated researchers to publish their data as scientific publications (as data papers [2] or as technical documentation to be uploaded in open data portals). Even though these papers include a number of the desired dimensions, they are written in natural language, and their structure and content are not fixed, making them challenging to study and analyze.

In this work, we present DataDoc Analyzer, a tool to analyze scientific dataset documentation by extracting the demanded dimensions and checking its level of completeness. We believe that our tool could help improve the explainability of scientific data by annotating the dataset with the extracted dimensions, assist data creators during the creation of datasets documentation, and be a helpful tool for empirical studies of the datasets in the ML field.

The architecture of the tool is composed of an ML pipeline that extracts and prepares the data from the documentation, builds a chain of prompts to be ingested by a large language model (currently GPT3.5), and then classifies the obtained answers using a fine-tuned version of BART to get the level of completeness. The tool comprises a web UI suited to test its capabilities and an API ready to be integrated into any data pipeline. The demonstration of the

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52021PC0206
[2] https://www.whitehouse.gov/ostp/ai-bill-of-rights/

tool's web UI is accessible as a HuggingFace space, and the code, together with the instructions to use it, are open source[3].

2 GUIDELINES TO DOCUMENT DATASET

The broad baseline for dataset documentation is stated clearly in Gebru et al.'s publication *Datasheets for Datasets* [4]. The concept of datasheets comes from the electronics field, where each component has an associated datasheet as documentation. *Datasheets for Datasets* and later publications in the area specify the dimensions that need to be defined for datasets intended for use in machine learning [4, 9, 1, 5]. Table 1 provides an overview of these dimensions, which are the focus of the tool's extraction approach.

The *Uses* dimension corresponds to the authors' stated design objectives, and we focus on extracting the purposes of the dataset, the gaps it is intended to address, and its recommended and non-recommended uses. Furthermore, we hope to infer the machine learning task for which the dataset was built, as well as the dataset's machine learning (ML) benchmarks, if this has been tested in any ML approach. *Contributors* refers to all participants in the dataset production, financing information, and the dataset's set of maintenance policies. The *Distribution* dimension contains information about the locations where the data can be accessible, the policies under which the dataset is published, and the dataset's deprecation policies. The *Composition* dimension applies to the file format, attributes, recommended data splits for training ML models, and pertinent dataset statistics.

In terms of data provenance, the *Gathering* dimensions correspond to information about how the data was gathered. This dimension's objective is to obtain a description of the process and infer its type (from a list of pre-defined types), information on the gathering team, the data source, the infrastructure used, and the process's localization. Furthermore, the *Annotation* dimensions focus on aspects of the dataset labeling process, such as the team annotating the data, the infrastructure employed, or the methodologies used to evaluate the labels. Finally, the *Social Concern*s dimension includes information regarding the potential effects of the data on society, such as biases, representativeness (for example, biased diagnosis), or data privacy concerns.

3 ARCHITECTURE

The tool's workflow is composed of three stages (see Figure 1). In what follows, we describe each stage.

3.1 Data preparation

The input of our extraction approach is the documentation accompanying the dataset. These documents, mainly made up of text, frequently appear in standard formats like PDF or HTML. In both cases, the text can be easily extracted. As part of the text, we also extract the content in the tables that could appear in the document. We have used GROBID [10] to extract the running text from PDF format and Tabula-py[4] to extract the tables.

[3]https://github.com/SOM-Research/DataDoc-Analyzer

[4]https://pypi.org/project/tabula-py/

Table 1: Target dimensions of the extraction approach

Dimensions		Target explanation
Uses	Design intentions	The ML tasks, the purposes, and the gaps the dataset intends to fill
	Recommendations	Identify the recommended and non-recommended uses
	ML Benchmarks	The ML approaches the dataset has been tested (if any)
Contributors	Authors	The authors of the dataset
	Funding	The funders and the funding information (grants, fudnder's type)
	Maintenance	Maintainers and maintenance policies (erratum, contribution, updates)
Distribution	Accessibility	The links where the data can be accessed
	Licenses	Legal condition of the dataset and the models trained with the data
	Deprecation policies	The deprecation plan for the dataset.
Composition	Data records	File composition and attribute identification
	Data splits	Recommended data splits
	Statistics and consistency rules	Relevant statistics pointed in the documentation
Gathering	Description & type identification	Description of the process and its categorization
	Team	Information about the type and demographics of the team
	Source & infrastructure	The source of the data and the infrastructure used to collect it
	Localization	Temporal and geographical localization of the data
Annotation	Description & type identification	Description of the process and its categorization
	Team	Information about the type and demographics of the team
	Infrastructure	The tools used to annotate the data
	Validation	Validation methods applied over the data
Social Concerns	Bias issues	Potential bias issues mentioned
	Representativeness or Sensitivity issues	Potential representative or sensitivity issues
	Privacy issues	Issues concerning privacy issues (p.e: anonymization)

Once the text is extracted, we split it into short chunks, between 200 and 300 words, aiming to respect the paragraph structure. Besides, to facilitate the comprehension of the content of the tables by the LLM, we have converted the tables to natural text explanations. To do so, we have built specific prompts with the paragraphs mentioning the table, the table, and an instruction to generate a natural language explanation of the table's content. Once we get the explanation, we treat it as any other paragraph of the running text.

Finally, we encode the paragraphs in a dense vector representation using GPT3.5 (*text-embedding-ada-002*), and we index them using FAISS [6] to perform semantic similarity in the following stage. The process is only done the first time the tool "sees" a specific document. Indeed, to avoid costs and increase the tool's response time, the results of the *data preparation* stage for each document are cached.

3.2 Dimension extraction

In this phase, for each target dimension (see Table 1), we have built a chain of prompts to be ingested by the LLM. The chains are composed of different types of prompts that aim to extract a specific dimensions while avoiding hallucinations issues.

Figure 2 shows an example of a chain that aims to extract the dataset's intended task. The first prompt is an example of an extractive prompt. It is composed of a specific *query* (the queries have been designed heuristically by the authors) and a set of *relevant passages* in the form of context. The following prompt is derived from the previous answer, together with a specific instruction asking to classify such answer into one of the particular ML categories[5].

The implementation of the chains is done via LangChain[6], and uses GPT3.5 (text-davinci-003) through the API service provided by its vendors. However, the tool is agnostic from the underlying LLM, and has also been tested using open-source models such as FLAN-UL2. Finally, to avoid costs and improve the tool's time response, we have parallelized, as far as possible, the requests to the LLM.

3.3 Post-processing

The final phase of the tool is meant to analyze the obtained answer from the LLM in order to evaluate as well its completeness. To do so, we have used a distilled version of BART [8] fine-tuned on the MultiNLI (MNLI) dataset [12] to perform zero-shot classification. For each dimension, we provide the model with a set of relevant categories (for instance, "Is there a localization" and "Is not there a localization"), and we ask the model to classify it. Then, we compile all the answers to generate a report, allowing the user to evaluate not only the concrete values but also the overall completeness of the documentation regarding the demanded dimensions.

[5]The list has been extracted from HuggingFace: https://huggingface.co/tasks
[6]https://github.com/hwchase17/langchain

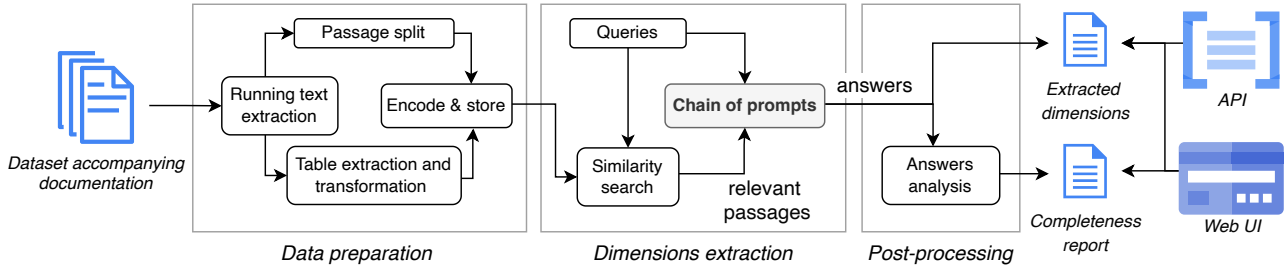

Figure 1: The workflow of the tool

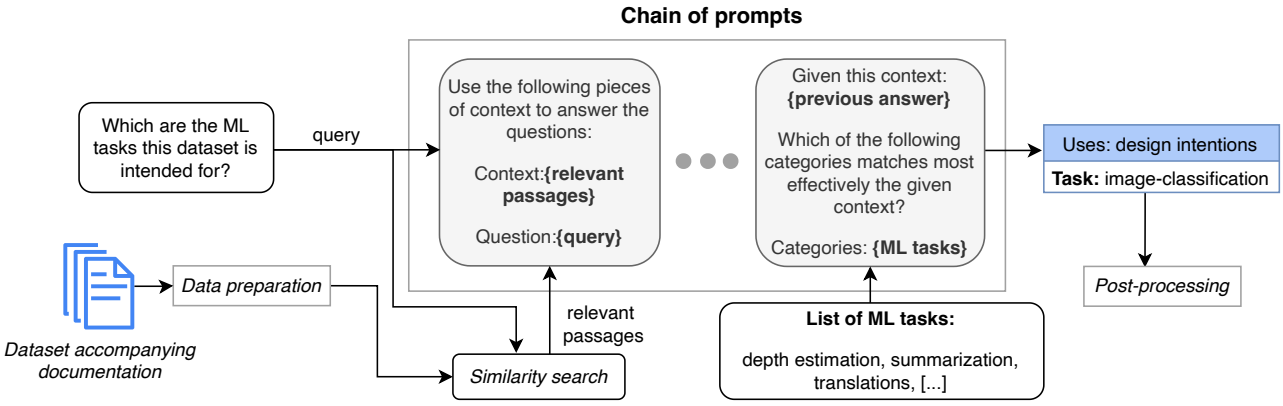

Figure 2: Example of a chain extracting the dataset intended task

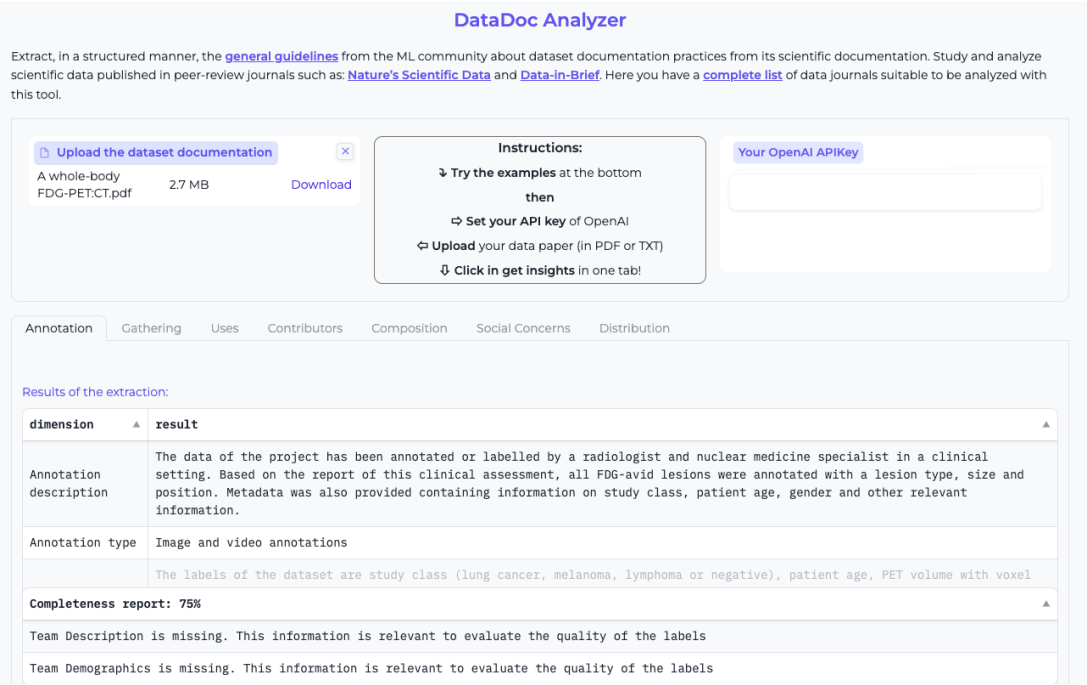

Figure 3: Web UI overview

4 TOOL USAGE

DataDoc offers two different user interfaces, a Web UI implemented with Gradio[7] intended to demonstrate the capabilities of the tool and suited to analyze a single document, and an API able to be integrated into any data processing pipeline. Figure 3 shows an overview of the Web UI, and a running demo of this one is published in HuggingFace spaces[8].

In the figure, we can see a set of simple instructions to use the tool's demo. First, users can try the examples at the bottom of the page to see the tool's capabilities in analyzing real datasets. The examples are two datasets, *A whole-body FDG-PET/CT Dataset with manually annotated Tumor Lesions*[9] and *DeepLontar dataset for handwritten Balinese character detection and syllable recognition on Lontar*[10] published from the Nature's journal Scientific Data.

To test other documents, following the instruction, the user needs to set up the *API key* and upload a .txt or .pdf file of a data paper. Once the document is uploaded, the user can go across each dimension to extract its information and completeness report.

Since the demo is suited for testing the tool's capabilities, we also provide an API with a set of endpoints that reproduce the tab's behavior but return the information in a JSON format ready to be ingested in any data processing pipeline. We build the API using FastAPI[11], the API comes with documentation and usage instructions found in the repository, and we provide a docker[12] image to facilitate its usage. Regarding response time, processing unseen documents takes between 50 and 60 seconds. For already-seen documents (data preparation stage is cached), times go down to between 20 and 25 seconds for each dimension.

5 CONCLUSIONS

In this work, we have presented DataDoc Analyzer, a tool for analyzing dataset scientific documentation. The tool is published through a Web UI for testing purposes, and as an API ready to be integrated in any ML pipeline. We believe that this tool could help to improve scientific dataset documentation, and be a helpful tool for dataset's empirical studies in the ML field.

The tool is powered by GPT.3.5, but it is LLM agnostic. Therefore, the set of emerging LLMs opens a path to explore the capabilities of open-source models, and to find a fine-tuned version of a smaller one, for cheaper and faster inference.

ACKNOWLEDGEMENTS

This research has been partially supported by the Spanish government (LOCOSS - PID2020-114615RB-I00), the AIDOaRt project, which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007350. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Austria, Czech Republic, Finland, France, Italy, and Spain. Jordi Cabot is supported by the Luxembourg National Research Fund (FNR) PEARL program, grant agreement 16544475.

[7] https://gradio.app/
[8] https://huggingface.co/spaces/JoanGiner/DataDoc_Analyzer
[9] https://www.nature.com/articles/s41597-022-01718-3
[10] https://www.nature.com/articles/s41597-022-01867-5
[11] https://fastapi.tiangolo.com/
[12] https://hub.docker.com/r/joangi/datadoc_analyzer

REFERENCES

[1] Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6, 587–604.

[2] Vishwas Chavan and Lyubomir Penev. 2011. The data paper: a mechanism to incentivize data publishing in biodiversity science. *BMC bioinformatics*, 12, 1–12.

[3] Sagar Bhimrao Gajbe, Amit Tiwari, Gopalji, and Ranjeet Kumar Singh. 2021. Evaluation and analysis of data management plan tools: a parametric approach. *Information Processing & Management*, 58, 3, 102480. DOI: https://doi.org/10.1016/j.ipm.2020.102480.

[4] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64, 12, 86–92.

[5] Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2023. A domain-specific language for describing machine learning datasets. *Journal of Computer Languages*, 101209. DOI: https://doi.org/10.1016/j.cola.2023.101209.

[6] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7, 3, 535–547.

[7] Agostina J. Larrazabal, Nicolás Nieto, Victoria Peterson, Diego H. Milone, and Enzo Ferrante. 2020. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, 117, 23, 12592–12594. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1919012117. DOI: 10.1073/pnas.1919012117.

[8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

[9] Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. 2021. Reusable templates and guides for documenting datasets and models for natural language processing and generation: a case study of the HuggingFace and GEM data and model cards. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics*. ACM, Online, 121–135.

[10] Laurent Romary and Patrice Lopez. 2015. GROBID - Information Extraction from Scientific Publications. *ERCIM News*. Scientific Data Sharing and Re-use 100, (Jan. 2015). https://inria.hal.science/hal-01673305.

[11] Leho Tedersoo et al. 2021. Data sharing practices and data availability upon request differ across scientific disciplines. *Scientific data*, 8, 1, 192.

[12] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*. Association for Computational Linguistics (ACL), 1112–1122.