Automated metadata extraction: challenges and opportunities

Tyler J. Skluzacek
Data Lifecycle and Scalable Workflows Group
Oak Ridge National Laboratory

Kyle Chard and Ian Foster
Department of Computer Science, University of Chicago
Data Science and Learning Division, Argonne National Laboratory

***Abstract*—Proper application of the FAIR data principles is what separates a vibrant data ecosystem, in which research data are frequently shared and reused, from a lifeless data graveyard. Automated metadata extraction systems have been proposed as a means of bolstering the findability, interoperability, and reusability of data repositories with little or no human intervention. These extraction systems mine metadata by crawling a repository and applying lightweight *extractors* that, for various types of file (e.g., image, CSV file), extract or synthesize relevant attributes. In practice, however, the automated creation of generally useful metadata is fraught with challenges. Data consumers may have different perspectives as to what metadata representations are useful, the standards for recording metadata tend to change over time, and the software model for processing updates can introduce unnecessary human and computational effort. Thus, generalizing extraction for a broad audience of data consumers is a difficult and relatively unsolved problem.**

In this work, we explore these challenges faced by extraction systems in the context of constructing our own extraction system for science data. We first define the metadata extraction problem and provide context to the issues faced in generalizing metadata. Additionally, we identify potential research directions to help alleviate many of these challenges for all automated extraction systems. Ultimately, this work represents a first step in designing ubiquitous metadata extraction systems that can maximize the value of research data while minimizing the human efforts required in doing so.

***Index Terms*—metadata extraction, automation, data mining, file forensics**

I. Introduction

Advances in science instrumentation and computation are enabling the generation of unprecedented volumes and varieties of data throughout the research data life cycle [1]. A single experiment can create a large, evolving research corpus containing many types of data: raw data; cleaned data; derived data; scripts, configurations, and lab notes employed in collecting and cleaning the data; notebooks analyzing cleaned data; and resulting published research artifacts (graphics, papers, presentations, and shared data appendices). Efforts to organize research results in order to make these data Findable, Accessible, Interoperable, and Reusable (FAIR) are continuously required throughout the data life cycle to record key provenance (context) and file attribute (content) metadata [2]. Unfortunately, such organization is oftentimes deferred to the end of the research project when key contextual details are lost [3]. Further, this challenge is amplified when multiple people contribute data to the research corpus, data consumers have conflicting perspectives as to how data should be organized, or if the research data are so large (i.e., millions of files, multiple petabytes) as to prohibit manual annotation. Thus, we see many repositories plummet from rich data ecosystems into dormant data graveyards.

A potential tool for resurrecting these data graveyards is an automated metadata extraction system that crawls file stores, opens each file, and orchestrates the application of one or more metadata *extractors*—lightweight functions that input data and output metadata attributes—to each file. To address the needs of science-specific data repositories, we created the open source Xtract extraction system [4]. While we have shown in prior work that Xtract is scalable relative to its counterparts with similar extractor models [5], we discovered multiple contexts in which the metadata were not generally useful without significant customization to the library of extractors. This led us to the challenging questions: (1) why is automatically mining generally useful metadata difficult, and (2) what future research can lead to more useful metadata outputs given fixed compute budgets? In this paper, we strive to both contextualize and answer each of these questions.

The remainder of this paper is as follows. §II introduces the goals and components of metadata extraction systems. §III describes the Xtract metadata extraction system for science. §IV illuminates the challenges of automated metadata extraction. §V outlines potential new research directions. §VI describes related metadata extraction systems. Finally, §VII presents future work and concluding remarks.

II. Metadata Extraction Systems

Metadata extraction is the orchestration and application of *extractors*—lightweight functions that input data of a predefined format and output descriptive metadata—to a data repository. These repositories generally encapsulate and store data as files containing various data *types*. We [4], and others [6], have shown that an individual file can contain

multiple types of data; for instance a single CSV file containing row-column data beneath a free text description of an experiment contains both free text and tabular data types. Metadata extraction begins by using an `isolate` method to demarcate data *regions*: contiguous, uniformly typed portions of a file. Each region maps to one or more extractors that specialize in processing data of that type. We call the method for determining eligible extractors for a region `select`. Finally, the `apply` method executes the extractor on the region and saves the metadata. We illustrate a simple algorithm for sequential metadata extraction in Algorithm 1. We claim that any computing system capable of automatically performing even trivial versions of `isolate`, `select`, and `apply` can be considered a *metadata extraction system*.

Algorithm 1 : Simple Sequential Metadata Extraction

Inputs:
$f \in F$: files in the repository
$e \in E$: extractors in extractor library

```
metadata = dict()
quality_metrics = dict()

for f in F:
    // Isolate regions
    regions = isolate(f)

    for region in regions:
        // Get all eligible extractors
        E* = select(E, region)

        for e in E*:
            // Use extractor to retrieve metadata
            metadata = apply(e, region)
            // Calculate quality metrics of metadata
            metrics = rate(metadata)

            metadata[f.path][e] = metadata
            quality_metrics[f.path][e] = metrics
return metadata
```

Metadata alone are often not enough to describe data in FAIR fashion; many data consumers may require methods to validate the metadata extraction system itself, or by extension, the *quality* of metadata it provides. These quality metrics should be calculated via the `rate` method. The metadata quality literature points to many ways in which metadata can be evaluated, for example:

- **Completeness:** the proportion of total possible metadata attributes mined from a file region [7].
- **Correctness:** the degree to which the values represented in the metadata are free from error [8].
- **Readability:** the degree to which the metadata outputs can be read by humans; a proxy measure for semantic searchability.

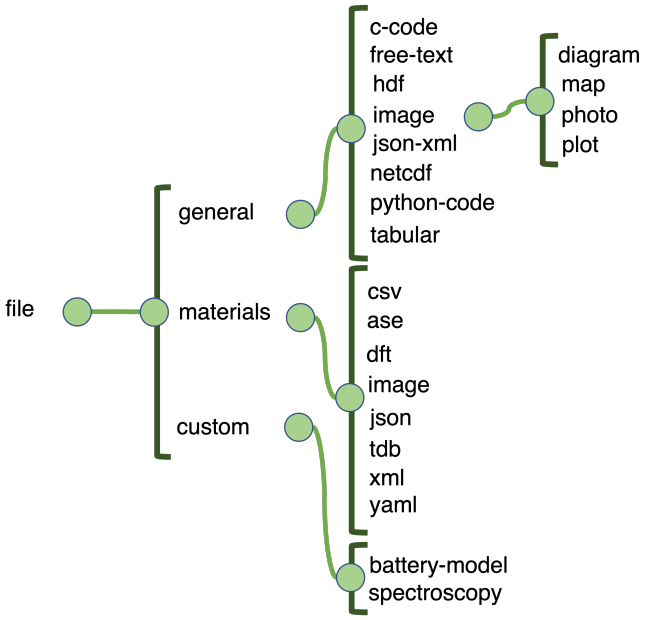


Fig. 1: Tree diagram of data types able to be processed by Xtract. Following the diagram from left-to-right leads to increasingly-specific file types, from all files being a *file* to a specific subsection of files being a particular subtype (e.g., *image.map*).

- **Entropy:** the degree to which metadata presents information that is different from other metadata [9].
- **Relevance:** the degree to which the metadata is useful to the context of its use [10].

In theory, the union of metadata with their associated quality metrics should allow data consumers to critically navigate vast data collections in order to accomplish a given use case.

III. Xtract

The Xtract metadata extraction system provides scalable extraction capabilities for large science repositories. To facilitate scalable execution across diverse research cyberinfrastructure, Xtract leverages a federated function as a service (FaaS) architecture to remotely orchestrate extractor execution at the edge. Xtract has been used to process over 500 million files (1 PB) in disciplines spanning climate, batteries, spectroscopy, materials, and beyond.

Xtract's extractor library reflects the data processing needs of these disciplines; users create and edit extractors to achieve metadata exhibiting high-enough quality metrics for their use cases. To submit or change extractors, users only need to abide by five broad design principles—that each metadata attribute is relevant and correct, and that the extractors are lightweight, modular, and able to flexibly adapt to minor schema differences between similar data regions. Users have created extractors capable of handling the 26 data types illustrated in Figure 1.

We next describe how Xtract performs each of the metadata extraction methods (as outlined in §II). `isolate` and

`select` are performed in tandem; lightweight statistical learning models determine the type(s) of a given file with high accuracy [11], and each file type automatically maps to one or more extractors. `apply` occurs by remotely invoking a FaaS function representing an extractor. Finally, `rate` occurs by applying Xtract's quality metric calculation library capable of computing metadata completeness, entropy, readability, and yield to the full corpus of metadata.

IV. Challenges

It is difficult to construct an automated metadata extraction system that robustly satisfies the evolving metadata quality needs of diverse users. In this section, we outline where the current state of research falls short by pointing to challenges discovered while building and evaluating Xtract. Specifically, we outline the challenges of managing divergent data perspectives, evolving quality standards, and a rapidly expanding pool of extractors.

A. Divergent data perspectives

Data consumers are unlikely to have identical preferences regarding the constitution of their metadata. We confirmed these misaligned preferences during a study of 6 active users of two large science repositories. In this study, we sought to determine whether, and to what extent, automatically extracted metadata promote science repository navigability.

Ideally, data consumers from similar science domains would have similar metadata preferences so that one user's extractor contributions provide research value to many. As part of this study, we conducted informational interviews with 3 active users of a battery data repository (Users 1, 2, and 3). We simply asked these users to list their desired metadata to be used as part of their research workflows, and uncovered stark differences in metadata preferences. Users 1 and 2 primarily wanted metadata that could be presented visually on a 2-dimensional graph, so that data consumers could "pull out quantities for specific parts of a voltage curve" (User 2). On the other hand, User 3 prioritized metadata attributes that promoted discovery so that they could "find data that are similar enough to treat with the same analytical technique". In this case, users desire different—but not conflicting—sources of metadata.

We also observed a situation where users desired conflicting metadata. Beyond users of the battery data repository, we also studied 3 active users of a spectroscopy repository. All 3 users (Users 4, 5, and 6) requested metadata about the brand of detector that generated a data set: Brand X or Brand Y. Each data set contained a "Detector" field containing "Brand X", "Brand Y", or nothing. In the case of an empty field, User 4 stated that the corresponding metadata attribute should read "Brand Y" due to it being a well-known bug with their system. User 6, on the other hand, stated that the attribute should read "unknown" since there is some non-negligible chance that the detector type is *actually unknown*. In this case, the extraction systems cannot please both users without compromising the metadata model.

B. Evolving metadata requirements

The value data consumers realize from data should decrease over time as users' needs evolve. Changes to user preferences can stem from new standards delineating how data should be indexed, the discovery of alternative uses for existing data sources, and the rise of new methods for computing metadata.

New metadata standards arise as science communities explore methods for making data more FAIR. We observed this firsthand while studying battery data repository users who wanted their metadata to adhere strictly to modern ontologies for indexing battery data [12]. The challenge here is obvious: every time an ontology changes, so must the extractors.

The discovery of alternative uses for existing data can also require changes to metadata. Fostering interoperability in bioscience requires all data consumers from all relevant domains to collaboratively characterize data for cross-disciplinary use [13]. For automated metadata extraction, representatives from each discipline must collaboratively design extractors for a given data type. Thus, as data consumers appear from new disciplines, the extractor library will likely need to change.

Finally, new methods for computing metadata attributes will lead to decreased utility of those computed using 'old' methods. In the same way that named-entity recognition (NER) enables users to classify and categorize entities in unstructured text, automated object recognition enables machines to detect and classify items in photographs and videos. When previously-inaccessible metadata suddenly become computable, extractors should be adjusted to include them, when useful.

Each of the aforementioned reasons for diminishing metadata satisfaction over time can only be resolved by overhauling the extractor library.

C. Extractor Library Growth

The long-term goal for metadata extraction systems is to have many users spanning diverse disciplines adding extractors and consuming metadata. As such, this uptick in activity could increase effort required by both humans and the system.

While in prior work we have shown that the invocation of individual metadata extractors is scalable, we simultaneously showed that extraneous extractor executions can waste significant resources when processing large data sets [5]. As new data perspectives and standards are introduced, updates to extractors will trigger their re-execution on corresponding files. As more extractors are added, the amount of duplicated work between extractors increases; all extractors must open and parse files, and some extractors might perform redundant computations. For example, separate extractors to compute the keywords and topic tags from free text data would both have to tokenize data before processing. These duplications of work, if run millions of times, can lead to significant wasted core hours on compute allocations.

When new data types are introduced, systems like Xtract and Tika [14] that use file type identification to identify data types must update their file type identification (FTI) tool and

re-execute on each file. In the case of Xtract, this means retraining a machine learning model each time a new data type is added and re-running the FTI tool on all files in the repository.

At current pace, the human effort in updating extractors will become more challenging; each extractor will get more complex as more people edit it. Users will need to ensure that their added capabilities do not overlap or break the capabilities created by others. Further, additional due diligence will be needed to determine whether an existing extractor can simply be updated or if an entirely new extractor is needed. Ideally, much of these software engineering steps should be hidden from the user so as to promote widespread adoption.

V. Potential Solutions

This section discusses promising research directions to better automate metadata extraction: highly modular extraction programs and multi-context metadata views.

A. Highly modular extraction programs

Extractors tend to have coarse responsibilities—they open files of a given type, isolate relevant data regions, and extract or synthesize multiple metadata attributes. As mentioned in §IV, the current extraction model creates undue effort required by both humans and computing systems. In the following, we explore decomposing the extraction logic of extractors and the `isolate`, `select`, `apply`, and `rate` functions into even-smaller functions called *microextractors*. We discuss how this model can be more computationally efficient and less burdensome for humans than existing approaches.

Preliminary guidelines for constructing microextractors are as follows:

- **Perform a minimally viable task on a well-defined data type.** These could include converting tabular columns to data series, converting JSON lists to data series, or computing specific values from a data series.
- **Possess a clear and editable dependency graph.** Outputs from one microextractor may be inputs to others. Having a clear dependency representation facilitates the use of leading workflow management strategies [15], [16] to orchestrate extraction. Additionally, methods for computing metadata can change over time. As such, the microextractor graph should be easily edited to accommodate adding, removing, and splitting tasks.
- **Support reasonable storage of relevant state.** Microextractors should checkpoint relevant metadata about their extraction so as to minimize redundant computation time as new extractors are added or updated.

Microextractors can be dynamically pipelined to perform all existing capabilities of current extractors. We present a potential simple microextractor workflow for handling files containing data series and/or text blobs in Figure 2.

Microextractor workflows, in theory, should require less human and computational effort to implement and maintain. Human effort is minimized as the metadata extraction logic is represented as a clear program flow between nodes of a graph (i.e., one should be able to identify duplicated work or an entry point for a new microextractor), the extraction logic of each microextractor is relatively simpler than that of current extractors (i.e., easier to understand, program, and test), and the mechanism for adding extractors requires simple graph manipulations.

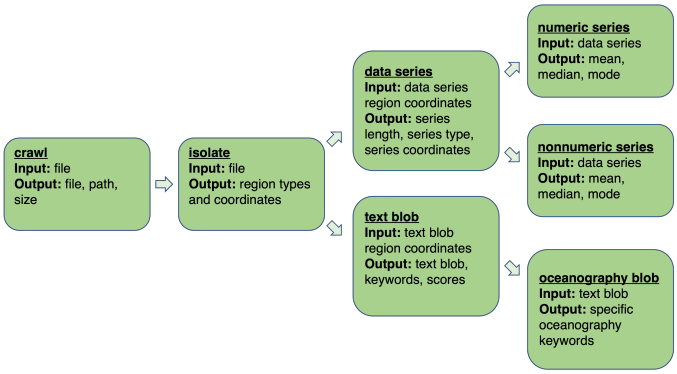


Fig. 2: A simple microextractor workflow with capabilities to handle files containing data series and/or text blobs.

Reducing redundant extractor logic enables only core computational capabilities to be executed on one's computational resources, thereby wasting fewer compute cycles. Updates to microextractor workflows can also benefit from strategically stored state of metadata extraction. Future research is needed to explore microextractor checkpointing strategies to minimize the time taken to update the metadata corpus subject to storage constraints.

B. Multi-context metadata views

Different data consumers may have unique metadata specifications, even when the metadata are generated from the same extractors. Consider a case where users from separate domains wish to parse a search index containing metadata for academic papers. The index is populated by using Xtract's `keyword` extractor that mines and ranks keywords from text documents. An ornithologist searching through this index for information about *gavia immer* (the common loon) would likely fail to be helped by less specific keywords. However, a cultural historian studying the influence of birds in popular media would likely not know what to do with a *gavia immer* tag, and might instead prefer to search for *bird*.

A multi-context view of this metadata would enable different data consumers to view data with personalized specificity levels. Specificity, in this case, broadly encompasses the definition of words, precision of values, or simply the presence of some attribute. Hierarchical models have been shown to improve search capabilities for images [17], text documents [18], and numeric data [19]. However, as a data consumer's context gets more specific, further research is needed to dynamically partition these hierarchies subject to both the context and user filters. Research in recommender systems [20] and assistive chatbots [21] could help dynami-

cally cater a consumer's metadata view based on their search history and feedback.

Metadata quality metrics should also account for variance in preferences between data consumers. To illustrate, consider the following hypothetical example. There are two data consumers *A* and *B*. A given metadata document contains 15 out of a possible 20 metadata attributes, for a completeness score of 75%. Fortunately, *A* needs 10 metadata attributes for their use case and all 10 are present. On the other hand, *B* also needs 10 metadata attributes for their use case, but 5 are missing. Thus, *A* has a subjective completeness of 100% while *B* has a subjective completeness of 50%. Further, it could be that one of the missing attributes is absolutely vital to *B*'s use case such that, without it, they cannot perform their use case. In that case, *B* experiences a subjective completeness of 0%. Thus, *B* would likely need to adjust one or more extractors in order to realize any value from the data. Therefore, the process of metadata extraction should include calculating and presenting *subjective* quality metrics, when possible, to reflect actual usefulness for each data consumer.

VI. Related Work

A number of systems have been developed to automatically extract and organize metadata from files. In the following, we review several such systems and compare them to Xtract.

Apache Tika [14] is an open source metadata extraction toolkit and extractor library originally constructed to support the Apache Nutch web crawler [22]. Tika has an extensible parser interface for developing custom parsers, and disparate science organizations have adopted it for extraction purposes. With its widespread adoption, Tika's community has developed a rich extractor suite. A limitation is that the choice of parsers to apply to a file is made primarily on the basis of MIME types, which are often misleading in scientific data sets, where for example MIME type 'text/plain' may be used for both tabular and free text files. The Tika libraries are also used by the GEMMS [23] metadata extraction system to identify parsers for extracting property, structure, and semantic metadata; thus GEMMS suffers from similar limitations to Tika when applied to scientific data.

Constellation [24] is a centralized metadata extraction and storage system that extracts entities—research groups, machines, experiments and files—from scientific data. It provides simple extractors for self-described hierarchical metadata, HPC logs, and file systems. Also relevant to Xtract is work on pay-as-you go information integration systems, which allow for incremental improvements to semantic mappings between data elements, as users decide that further investment in data analysis is required [25].

The Big Data Quality Control (BDQC) Framework [26] is a centralized bulk metadata extraction system that enables search across large collections of biomedical data. BDQC is similar to Xtract in that it performs systematic 'bulk' sweeps across heterogeneous data and introspects files without regard to their meaning (domain-blind analysis), but with the primary goal of identifying anomalies. BDQC employs a pipeline of extractors to derive the properties of imaging, genomics, and clinical data. While BDQC is implemented as a standalone system, the approach taken would be similarly viable in a future microextractor implementation of Xtract.

VII. Conclusion

Automated metadata extraction has great potential for helping users organize and navigate their research data repositories. While many have designed extraction systems that have provided significant research value, there are still unsolved challenges in making their metadata outputs broadly useful. Divergent metadata perspectives of data consumers, evolving metadata standards, and extraction systems' rigid software components create complexity in creating, updating, and evaluating metadata. To address these issues, we present potential solutions for metadata extraction systems to simultaneously support the needs of both data consumers and the underlying compute resources.

Future work will explore metadata representations capable of curating data to a diversity of user perspectives as well as a modular microextractor model capable of handling extractor updates from each of these perspectives. Through these efforts, we strive to realize a future where metadata extraction systems create generally useful metadata, are user-friendly, and furthermore, are ubiquitously harnessed to improve the FAIRness of data collections.

References

[1] F. Berman, R. Rutenbar, B. Hailpern, H. Christensen, S. Davidson, D. Estrin, M. Franklin, M. Martonosi, P. Raghavan, V. Stodden *et al.*, "Realizing the potential of data science," *Communications of the ACM*, vol. 61, no. 4, pp. 67–72, 2018.

[2] W. P. Dempsey, I. Foster, S. Fraser, and C. Kesselman, "Sharing begins at home: How continuous and ubiquitous FAIRness can enhance research productivity and data reuse," *Harvard Data Science Review*, vol. 4, no. 3, 2022.

[3] L. Tedersoo, R. Küngas, E. Oras, K. Köster, H. Eenmaa, Ä. Leijen, M. Pedaste, M. Raju, A. Astapova, H. Lukner *et al.*, "Data sharing practices and data availability upon request differ across scientific disciplines," *Scientific data*, vol. 8, no. 1, pp. 1–11, 2021.

[4] T. J. Skluzacek, "Automated metadata extraction can make data swamps more navigable," Ph.D. dissertation, The University of Chicago, 2022.

[5] T. J. Skluzacek, R. Wong, Z. Li, R. Chard, K. Chard, and I. Foster, "A serverless framework for distributed bulk metadata extraction," in *30th International Symposium on High-Performance Parallel and Distributed Computing*, 2021, pp. 7–18.

[6] R. F. Erbacher and J. Mulholland, "Identification and localization of data types within large-scale file systems," in *Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'07)*. IEEE, 2007, pp. 55–70.

[7] M. Margaritopoulos, T. Margaritopoulos, I. Mavridis, and A. Manitsaris, "Quantifying and measuring metadata completeness," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 4, pp. 724–737, 2012.

[8] N. Palavitsinis, N. Manouselis, and S. Sánchez-Alonso, "Metadata quality issues in learning repositories," 2013.

[9] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[10] T. Margaritopoulos, M. Margaritopoulos, I. Mavridis, and A. Manitsaris, "A conceptual framework for metadata quality assessment." in *Dublin Core Conference*, 2008, pp. 104–113.

[11] T. J. Skluzacek, M. Chen, E. Hsu, K. Chard, and I. Foster, "Models and metrics for mining meaningful metadata," in *International Conference on Computational Science*. Springer, 2022, pp. 417–430.

[12] S. Clark, F. L. Bleken, S. Stier, E. Flores, C. W. Andersen, M. Marcinek, A. Szczesna-Chrzan, M. Gaberscek, M. R. Palacin, M. Uhrin *et al.*, "Toward a unified description of battery data," *Advanced Energy Materials*, vol. 12, no. 17, p. 2102702, 2022.

[13] S.-A. Sansone, P. Rocca-Serra, D. Field, E. Maguire, C. Taylor, O. Hofmann, H. Fang, S. Neumann, W. Tong, L. Amaral-Zettler *et al.*, "Toward interoperable bioscience data," *Nature genetics*, vol. 44, no. 2, pp. 121–126, 2012.

[14] C. Mattmann and J. Zitting, *Tika in Action*. USA: Manning Publications Co., 2011.

[15] Y. N. Babuji, K. Chard, I. T. Foster, D. S. Katz, M. Wilde, A. Woodard, and J. M. Wozniak, "Parsl: Scalable parallel scripting in python." in *IWSG*, 2018.

[16] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. F. Da Silva, M. Livny *et al.*, "Pegasus, a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, pp. 17–35, 2015.

[17] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of www image search results using visual, textual and link information," in *12th Annual ACM International Conference on Multimedia*, 2004, pp. 952–959.

[18] D. Kuang, X. Li, and C. X. Ling, "A new search engine integrating hierarchical browsing and keyword search," in *22nd International Joint Conference on Artificial Intelligence*, 2011.

[19] D. Hoang, B. Summa, H. Bhatia, P. Lindstrom, P. Klacansky, W. Usher, P.-T. Bremer, and V. Pascucci, "Efficient and flexible hierarchical data layouts for a unified encoding of scalar field precision and resolution," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 603–613, 2020.

[20] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics reports*, vol. 519, no. 1, pp. 1–49, 2012.

[21] X. Liu, W. Xue, Q. Su, W. Nie, and W. Peng, "metacat: A metadata-based task-oriented chatbot annotation tool," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, 2020, pp. 20–25.

[22] A. Nutch, "Apache nutch," 2014.

[23] C. Quix, R. Hai, and I. Vatov, "GEMMS: A generic and extensible metadata management system for data lakes," in *CAiSE Forum*. Ljubljana, Slovenia: Springer, 2016, pp. 129–136.

[24] S. S. Vazhkudai, J. Harney, R. Gunasekaran, D. Stansberry, S. Lim, T. Barron, A. Nash, and A. Ramanathan, "Constellation: A science graph network for scalable data and knowledge discovery in extreme-scale scientific collaborations," in *IEEE Internationial Conference on Big Data*. Washington D.C., USA: IEEE, 2016, pp. 3052–3061.

[25] A. Das Sarma, X. Dong, and A. Halevy, "Bootstrapping pay-as-you-go data integration systems," in *ACM SIGMOD International Conference on Management of Data*. Vancouver, Canada: ACM, 2008, pp. 861–874.

[26] E. W. Deutsch, R. Kramer, J. Ames, A. Bauman, D. S. Campbell, K. Chard, K. Clark, M. D'Arcy, I. D. Dinov, R. Donovan, I. Foster, B. D. Heavner, L. E. Hood, C. Kesselman, R. Madduri, H. Mi, A. Muruganujan, J. Pa, N. D. Price, M. Robinson, F. Sepehrband, A. W. Toga, J. V. Horn, L. Zhao, and G. Glusman, "BDQC: a general-purpose analytics tool for domain-blind validation of big data," *bioRxiv*, vol. 2018, no. 1, p. 258822, 2018.