Draining the Data Swamp: A Similarity-based Approach

Will Brackenbury, Rui Liu, Mainack Mondal,
Aaron J. Elmore, Blase Ur, Kyle Chard, Michael J. Franklin
University of Chicago
{wbrackenbury,ruiliu,mainack,aelmore,blase,chard,mjfranklin}@uchicago.edu

ABSTRACT

While hierarchical namespaces such as filesystems and repositories have long been used to organize data, the rapid increase in data production places increasing strain on users who wish to make use of the data. So called "data lakes" embrace the storage of data in its natural form, integrating and organizing in a Pay-as-you-go fashion. While this model defers the upfront cost of integration, the result is that data is unusable for discovery or analysis until it is processed. Thus, data scientists are forced to spend significant time and energy on mundane tasks such as data discovery, cleaning, integration, and management – when this is neglected, "data lakes" become "data swamps."

Prior work suggests that pure computational methods for resolving issues with the data discovery and management components are insufficient. Here, we provide evidence to confirm this hypothesis, showing that methods such as automated file clustering are unable to extract the necessary features from repositories to provide useful information to end-user data scientists, or make effective data management decisions on their behalf. We argue that the combination of frameworks for specifying file similarity and human-in-the-loop interaction is needed to aid automated organization. We propose an initial step here, classifying several dimensions by which items may be considered similar: the data, its origin, and its current characteristics. We initially consider this model in the context of identifying data that can be integrated or managed collectively. We additionally explore how current methods can be used to automate decision making using real-world data repository and file systems, and suggest how an online user study could be developed to further validate this hypothesis.

ACM Reference format:
Will Brackenbury, Rui Liu, Mainack Mondal, Aaron J. Elmore, Blase Ur, Kyle Chard, Michael J. Franklin . 2018. Draining the Data Swamp: A Similarity-based Approach. In *Proceedings of Workshop on Human-In-the-Loop Data Analytics, Houston, TX, USA, June 10, 2018 (HILDA'18),* 8 pages.
https://doi.org/10.1145/3209900.3209911

1 INTRODUCTION

It has been nearly a decade since the widespread adoption of data lakes. This change in approach to data management, from up-front data integration to a Pay-as-you-approach [19, 29], offers multiple advantages, namely flexibility, speed to insight, and simplicity of implementation. The downsides, however, have become more prevalent as the years have passed. The flexibility of data lakes becomes a hindrance at scale when the heterogeneity of the data does not allow for unified querying capabilities. The speed to insight is lost when requirements change and new datasets must be discovered, cleaned, and integrated into an analytics task. Even the simplicity of implementation becomes problematic when the low bar for implementation encourages lax data model design and poor metadata management. Data lakes, when mired in these complications, can then be more accurately labeled as "data swamps" [16].

This phenomenon is best illustrated by example: take the analytics pipeline at a large hospital network. When the network's pipeline was initiated, it may have consisted of processing a single set of unclean patient visitation reports from one hospital in the network. As time passes, this may grow to the same set of reports from multiple hospitals in the network, then to different sets of reports from these hospitals, then to reports from outpatient clinics, and so on. Each of these items requires a new *ad-hoc* solution to process. The overhead of discovering, cleaning, integrating, and managing all this data, which was once trivial, has become an imposition.

Providing pure, generalizable, computational methods to resolve the challenges posed by each of these components has been an area of research focus for some time now. Researchers have made significant progress in many sub-areas, such as data cleaning [20, 23, 31, 36] and data integration [5, 12, 19, 34, 35], and other metadata management tasks [17, 18]. Some items, however, remain elusive. While many critical problems must be addressed to help realize the vision of data lakes, we believe that data discovery through similarity (e.g. show me more like this) is a critical first step that is not addressed by existing tools. Automated systems for data discovery exist for specific environments, but in general are unable to provide users with successful dataset recommendations [14]. Automated data management has had success with semantic tagging solutions [25, 30, 32], but provides limited utility without an infeasibly large amount of user input.

This incompleteness in data discovery and data management suggests that pure computational solutions do not provide the efficacy required for these applications. We take this perspective here: we seek solutions that require minimal expert user input to produce results that are usable in practice. This idea of minimizing the needed amount of human-in-the-loop input is a familiar thread in several arenas [33], but we believe that it is yet to be applied in this domain.

This paper serves as a first step in a project to build a series of tools to aid in the taming of data lakes; through user studies we plan to evaluate our hypothesis on the benefits of similarity based tools for a variety of tasks, such as discovery, management, security, and

HILDA'18, June 10, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-5827-9/18/06…$15.00
https://doi.org/10.1145/3209900.3209911

organization. In this paper, we provide evidence demonstrating the difficulty of using pure computational solutions for data discovery and data management. Additionally, we propose a human-in-the-loop methodology for these tasks that collects user feedback both proactively and retroactively along the dimensions of the data itself, its origin, and the current characteristics in order to identify data that could be integrated or managed similarly. Lastly, we provide a basis for this methodology, and note suggestions for future work in conducting an online user study and developing tools founded on this methodology before concluding.

2 FRAMEWORK OF DATA SIMILARITY

We first define necessary terminology before proposing our framework of data similarity.

2.1 Definitions

We define **data discovery** to be the process by which an end-user discovers data relevant to a conceptual research question or query. For example, one can consider an employee at a major financial firm performing an experiment to predict how potential customers in a new market might respond to the opening of a new bank branch. If the employee starts with only the area's census demographics, they would likely be interested in a dataset that contains information on the financial status of those customers. A tool that can both identify and join related datasets would significantly reduce the time currently required for discovery. Accomplishing tasks of this type most often consists of identifying related terms based on how often they appear together, or identifying similarity between the schemas of two datasets.

In contrast, we define **data management** to be the task of appropriately organizing and maintaining existing content. This includes tasks such as duplicating items identified as important, encrypting files identified as sensitive, deleting superfluous files, and compressing or archiving files identified as unimportant. For example, one can consider a staff researcher at a pharmaceutical company who runs drug trials fairly often. As part of FDA regulations, they must securely store the data and maintain an audit trail. If they make local copies of the data for one-off statistical analyses, however, these copies can inadvertently become lost in the organization's data swamp. This worry is compounded by typical organizational structures; the swamp comprises data from data creators across multiple teams, each with domain-specific knowledge.

2.2 Proposed Framework

If an employee comes across a file in the data swamp and wishes to encrypt both that file and similarly sensitive files, they currently need to do so manually. If they wish to discover other data from the same project or on the same topic and join that data, they again must do so manually. A tool that could determine file similarity based on expected sensitivity, topic, or provenance would enable the employee to quickly scale this remediation. To that end, we propose a framework of data similarity, shown in Table 1. Below, we describe the three branches of this framework and define the ten aspects of data similarity that these branches encompass:

1) The Data Itself. The types of similarity that are most straightforward to measure are the characteristics of the data itself. For instance, data lives in at least one **location**. Data can be similar to other data in its relative location on disk, the contextual clues provided in the tokens of the data's directory path, which database data resides in, and which systems contain backup or derivative copies of data.

Table 1: Our proposed framework of data similarity contains three high-level categories capturing ten aspects of in which data can be similar to other data.

The Data Itself
• Location
• Metadata
• Contents
• Topic
Origin
• Provenance
• Reliability
• Initial Purpose
Current Characteristics
• Immediate Utility
• Retention Importance
• Sensitivity

Data can also be similar to other data in terms of its **metadata**, encompassing attributes from the size of the data to the extension to the time it was created or last modified. The contextual clues suggested in the name of the file are also valuable metadata, and files may be similar to each other in terms of file name (e.g., two files called "readme").

The **contents** of files or databases are an important way in which they may be similar to each other. Files whose contents on the level of bits are nearly identical (i.e., have a small edit distance) may be modifications or derivatives.

When we consider similarity in terms of contents, one can focus on similarities that are on the level of bits and therefore are easily machine-identifiable. However, files that superficially seem dissimilar on the level of bits may nonetheless be very similar in contents. For instance, the translation of a document into another language or a photo of the same object in vastly different lighting results in two files that may look very different based on naive notions of similarity, yet would likely be considered similar to a human. We conceive of this more complex notion of similarity in contents to be applicable whenever there is a direct mapping between the contents of two instances of data, yet an exceedingly complex computational model describes this translation.

Finally, data can be similar to other data in terms of **topic**, though this relationship may or may not be easily detectable computationally. Different instances of data about the earth's climate, even if from completely different sources and completely different experiments, is nonetheless similar in terms of topic. Similarly, cartoon drawings of dogs and photographs of dogs can likely be said to be similar in terms of topic.

2) Origin. Data can also be similar to other data based on its origin. First, the **provenance** can be the same, meaning that the data were created at similar times, in similar locations, and/or by related people or tools. For example, a short story written by one author

at a writer's retreat and a short poem written by another at the same retreat would be similar along this dimension. Notably, data originating from the same conceptual project, same authors, or from the same time period can be argued to have the same provenance, albeit in subtly different ways.

The **reliability** of data, or degree that a viewer can trust in its correctness, also derives from its origin. Data can be unreliable from its initial origin. This occurs in cases of data collected using faulty or poorly calibrated scientific instruments, information created by an author of low repute, or potentially even data where the source is unknown and therefore cannot be fully trusted. Data can also be made unreliable through subsequent transport and processing, including by imperfect data cleaning, lossy compression, improper subsetting, unexpected encoding, or undocumented derivation from a different data set.

Data can also be similar in the **initial purpose** for which it was created. Instances of data that differ in form may have been collected toward the same initial goal. For example, a receipt and a W-2 form that were both created for tax purposes would be similar along this dimension.

3) Current Characteristics. While the previous dimension focused on the origin of the data, the third dimension instead focuses on current uses for that data. First, one can consider the **immediate utility** of data. Over time, data might become more or less important. For instance, data that is highly relevant for a certain scientific experiment might be crucially important until a PI determines that the technique is not promising, at which point the data loses utility moving forward. In contrast, measurements of a particular distant galaxy taken years prior as part of a broad probe might suddenly become highly valuable and useful if interesting phenomena are later observed in that galaxy and scientists wish to characterize longitudinal changes.

Regardless of its utility past or present, data might differ or be similar in **retention importance**. Some types of data, such as family photos, are important to retain out of nostalgia. Other types of data, such as scientific data sets collected at great effort or cost but of no immediate utility, are important to retain in case they later become useful. Other data might have a legal obligation for retention, such as data collected during drug trials or financial records. In contrast, other data might have obligations for deletion, such as data a scientist has promised human subjects would be deleted at the conclusion of an experiment.

Finally, data can be similar to other data in terms of **sensitivity**. For instance, an electronic health record and a photocopy of a personal tax return might be similar in sensitivity to the data subject despite differing substantially in form.

2.3 Applying this Framework of Data Similarity

Many tasks relevant to draining data swamps can be facilitated by enabling the humans in the loop to express tasks based on one or more of the dimensions of similarity in our framework. For example, if a user is deleting a file or dropping a database table deemed to have no current or future relevance, that user might want to issue a command that the system delete other files from the same project. As a second example, a medical researcher might discover personally identifiable information in electronic health records on a backup system and then wish to audit the rest of that backup system to encrypt similarly sensitive data.

Current interfaces do not enable humans in the loop to express similarity with the rich vocabulary encapsulated by our proposed framework, rendering data management highly labor intensive for humans, particularly in cases involving retrospective management of large data swamps containing data from many users. Unfortunately, implementing this rich vocabulary for expressing similarity is far from trivial. We posit that data may be similar to data along some dimensions of this framework, but not others. That data can be similar in some dimensions, but not others, means that pure computation alone cannot fully solve this problem, yet we imagine that a human in the loop in concert with computation will enable such expressive HILDA interfaces. Minimizing the burden for human users is especially important because many retrospective data-management tasks cannot be crowdsourced both due to concerns about security and privacy and because crowdworkers likely lack the domain knowledge necessary to complete these tasks.

The process of human-in-the-loop data discovery is similarly handicapped by the inability for users to specify actions using the dimensions of this framework, as well as by the inability of systems to suggest other data sets that are similar in such ways. For instance, while working with a given dataset, a user might want to search for data of similar origin, or that is similarly reliable, or that is about a similar topic. Moving towards this vision is the basis for a long-term research agenda.

Moving forward, we hypothesize that it is possible to build a multidimensional similarity ranking along the dimensions of our framework with an active learning strategy that leverages schema similarity, file content similarity, and metadata similarity to guide initial tuple selection. Given limited, iterative feedback, the method could then provide recommendations of similar files when users request assistance from this tool in data discovery or data management, such that users can either join together similar files or perform other management actions on files that are similar along a specified framework dimension.

3 EXPERIMENTS SUMMARY

In order to validate the hypothesis that computational methods alone are insufficient to address the challenges of data discovery and data management in a data swamp, we use two corpora to perform experiments.

The first corpus is the Google WebTables dataset, a dataset consisting of 1B+ data records, extracted from several million web tables. This dataset was obtained from the Web Data Commons [1] that used multiple heuristics to identify relational tables on the web [11]. The second of these is a file system dump from the Carbon Dioxide Information Analysis Center (CDIAC), consisting of ~ 0.5M files that contain environmental science data. This second dataset is particularly noteworthy because it has several attractive characteristics that make it more like a data lake than a data swamp. Its directory structure is well-organized, files have informative names with associated metadata, and many of the tabular

[1] http://webdatacommons.org/webtables/

files are based on schemas are simple to extract. Therefore, experimental results on this corpus should perform well, and if they do not, this lends credence to the hypothesis that naive approaches are insufficient to address the challenges of data discovery and data management in data swamps.

We use these datasets to conduct two experiments. The first of which, is a pair of experiments using a schema-completion tool based on WebTables [11] to identify data with similar schemas. Here we define a task in which an end-user wants to locate datasets with schemas that can augment the schema of a given input dataset. Our second experiment investigates our ability to automatically group files that are considered similar based only on their metadata. We use the CDIAC dataset and compare our automatically created groupings with the manually assigned groupings derived from file system structure (i.e., namespace proximity).

4 SCHEMA SIMILARITY EXPERIMENTS

To evaluate the potential for pure computational solutions to data discovery challenges, we investigate our ability to automatically associate schemas. To do so, we implemented a schema completion tool, based on Google's WebTables [3, 11]. Given an input dataset of tables, we randomly remove 20-30% of the attributes from these tables , and pose the partial tables as input to the aforementioned schema completion tool. We evaluate the effectiveness of the tool based on whether it is able to impute the missing schema.

We evaluate this tool on both the original WebTables corpus and the CDIAC repository. We limit the number of tables considered (~100K tables) from these corpora to represent realistic file system or repository sizes and to make our experiments computationally tractable. Since the key objective of the experiment is to evaluate the performance of schema matching and discovery, we focus on the accuracy of schema auto-completion and attribute synonym identification.

Figure 1 illustrates our current implementation, which contains three major components: data extraction, schema exploration, and schema generation. Data extraction is responsible for reading the structured tabular data and eliminating formatting issues. Although WebTables relies on the huge corpus to generate accurate results, this pre-cleaning process would neither hurt nor improve the final results, given that our system cannot calculate the similarity between the words and schemas if the words cannot be recognized. Schema exploration aims to model the data relations and impute synonyms, based on the assumption that words often appearing in the same relations may be synonymous. Each explored schema is assigned a score: the higher the score, the greater probability that the schema has been inferred correctly. Finally, schema generation outputs all the explored schema and ranks them with the final scores from the previous step.

Evaluation Metrics: The output of the schema completion tool is: 1) for each table a ranked list of tables with schemas that are considered similar; and 2) for each word a ranked list of synonyms. The output statistics of running an experiment with different configurations can be used to evaluate the schema completion tool against two important performance metrics.

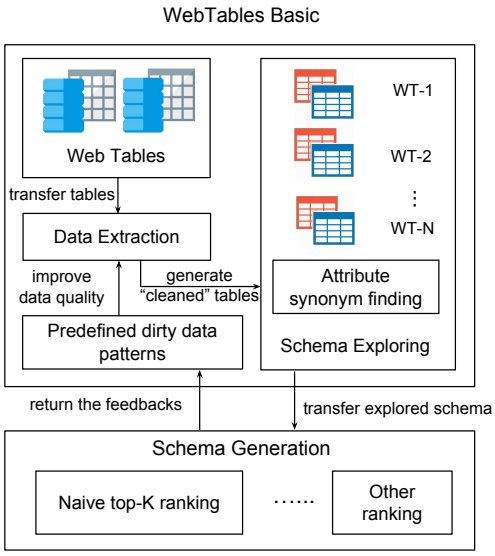

Figure 1: The architecture of our implementation.

- Accuracy: measured as the number of correct schema completions within the top-k results, given different k and input data size.
- Scalability: measured as the change in accuracy when increasing the input data size.

4.1 WebTables Corpus

Dataset: To reduce the complexity of our experiments we create a small-scale input dataset by randomly sampling the entire WebTables corpus. The resulting dataset contains 10,000 HTML tables from the Web. We randomly remove attributes for evaluation.

Preliminary Results: Fig. 2 demonstrates the difficulty of completing schema given minimal input data. Since the key idea of WebTables is to exploit probability to complete schema, the small-scale input dataset significantly reduces performance. Even in the Top-20 case (which is impractical in most real-world scenarios), the average accuracy is around 20%, insufficient for data discovery tasks. We also evaluate the synonym identification function, which is useful in schema exploration. These results are presented in Table 2. In the ideal case, our tool would correctly identify synonyms in the Top-1 instance. Unfortunately, the results are rarely correct in this case (2%). Even in relaxed cases, such as Top-5, Top-10, Top-15, and Top-20, our tool cannot identify the correct synonyms. Thus, limited input data size significantly affects the accuracy of the tool.

4.2 CDIAC Corpus

Dataset: We also evaluate WebTables on a dataset from CDIAC which contains a large collection of environmental science data. To make an apples-to-apples comparison, we uniformly sampled the whole dataset and pre-processed the data to obtain 100K tables (around 10GB). Following the same process as for the WebTables corpus, we randomly selected several schemas and removed some attributes, then queried the schema completion tool to determine if it was possible to accurately complete them.

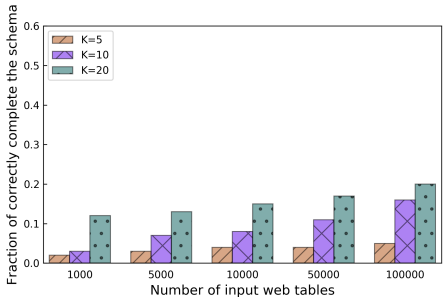

Figure 2: Fraction of correctly completed schemas

	Top-1	Top-5	Top-10	Top-15	Top-20
title	0.05	0.32	0.38	0.43	0.47
flag	0	0.02	0.08	0.03	0.09
location	0.02	0.05	0.09	0.12	0.13
student	0	0.08	0.12	0.2	0.23
username	0.04	0.13	0.29	0.34	0.35
report	0	0.01	0.04	0.07	0.09
AVG	≈0.02	≈0.1	≈0.16	≈0.17	≈0.23

Table 2: Fraction of correctly identified synonyms in schema exploration based on 10000 web tables

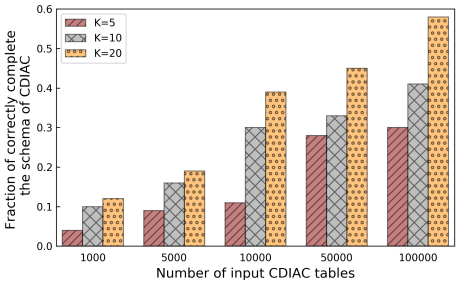

Figure 3: Fraction of correctly completing schemas of CDIAC

Preliminary Results: The results are shown in Fig. 3. These results are superior to those on HTML tables. We hypothesize that this is due to two reasons. First, CDIAC focuses on environmental data and therefore the data is inherently similar from a macro perspective. Second, most tables in CDIAC have similar layouts and content, perhaps as they were created by the same person or the same tool. For instance, there are many tables that report carbon levels in the Indian Ocean at different times. These tables use the same layout with the same attribute names. In this case, our tool performs well (e.g., synonyms can be found easily); however for some common attribute names (e.g., data or time) it is still difficult to complete them. In general, Fig 3 demonstrates that WebTables performs well if the given data are relevant and their quality are high. However, this is an almost ideal scenario and is not likely to be common

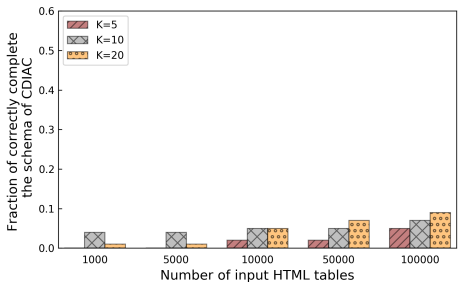

Figure 4: Fraction of correctly completing schemas of CDIAC using HTML Tables

in real-world scenarios. In future work we plan to explore if such differences hold between enterprise lakes and scientific lakes.

To evaluate our thesis that such tasks are more difficult in scenarios in which little is known about the specific context we evaluate schema completion on the CDIAC dataset when trained on the WebTables dataset. The results are presented in Fig. 4. Even for the most relaxed scenario, i.e., top-20, the accuracy is only 10%. In most cases, our approach didn't work at all. These results highlight the difficulty establishing similarity with little context known.

5 FILE SYSTEM METADATA EXPERIMENTS

We now turn to the question of evaluating file similarity using groupings derived from available file system metadata. To do so, we estimate the Jaccard similarity between file paths using a combination of MinHash [9] and Locality-Sensitive Hashing (LSH) [1]. We then compare these clusters against the logical groupings established by the directory structure. We evaluated these techniques on the CDIAC dataset with ~ 0.5M files in the repository. Our results are shown in Table 3. For this experiment, we conducted scans over different subsets of the total files, varying the number of files examined, as well as the permutations for MinHash, given that greater numbers of permutations increase the accuracy of MinHash in estimating the Jaccard similarity. We then estimated the Jaccard similarity of the file path for each file using MinHash, and bucketed it via LSH to group together files that had similarity greater than 50%. We iterated through the traversed files, and measured the percentage of similar items in each grouping that were also located in the same directory. We report the mean value of this percentage. For example, when 100 files in total were examined, and MinHash evaluated Jaccard similarity using 64 permutations, on average only 16% of the files in a given file's similarity grouping lay outside of the given file's directory.

The results align roughly with expectations: the percentage of files that lie outside of the directory grows as more files are examined. It is also notable that as the MinHash permutations increase, the percentage of files in a grouping that lie outside of the directory decreases, suggesting that low-permutation MinHashes overestimate similarity.

In a practical sense, though, if 85% of similar files (~ 0.5M files and 256 MinHash permutations) lie outside of the directory, then attempting to manage files in the same directory similarly will miss

a significant portion of the files that should be managed similarly. This would mean that such a data management approach would be highly ineffective, and would require significant additional input from the user in order to address the root challenge. Naturally, this is a very rough estimate, and state-of-the-art methods can certainly achieve higher accuracy. The particularities of the CDIAC dataset, however, make this result more surprising. Given its apparent grouping of similar files in similar directories, this suggests that even in clean repositories such as this one, significant additional explanatory power is required. This shows the greater impact of this initial result, and demonstrates that human computation could be necessary in order to increase the accuracy to a sufficient threshold.

		MinHash Permutations		
		64	**128**	**256**
Files Examined	$\mathbf{10^2}$	0.16	0.16	0.12
	$\mathbf{10^3}$	0.36	0.36	0.36
	$\mathbf{10^4}$	0.88	0.88	0.88
	$\mathbf{10^5}$	0.93	0.90	0.85
	$\mathbf{10^6}$	0.93	0.91	0.85

Table 3: Accuracy of Jaccard similarity comparison on filepaths

6 RELATED WORK

The taming of data swamps is comprised of four main components. We review existing research along each of these components.

Data discovery: Prior work studied the process of data discovery, often in specific contexts. Deng et al. [14] presented effective data discovery techniques for polystore systems, where data may reside in heterogeneous databases that support SQL queries. Balakrishnan et al. [3] designed a system for additional dataset recommendations by leveraging a relational-based corpus of 100M+ HTML tables collected from the web. Halevy et al. [17] designed and implemented a full data discovery system at Google to cluster datasets, annotate datasets, and identify relationships between datasets. Their system leveraged tightly coupled information about the generating code (from Google's infrastructure). Unfortunately, data swamps might not be polystore systems and might not contain relational-based data or well structured information about the generating code; Thus these techniques are largely insufficient in a general scenario. Furthermore, these existing systems mostly overlooked the need for a human-in-the-loop component in data discovery. Recently, Hellerstein et al. [18] touched upon the organization of both machine and user generated metadata to provide data context. However they did not explore the necessity and scope of exact human-in-the-loop mechanisms for data discovery in real-world data swamps, we aim to bridge that gap. Other related projects explore discovering attributes, synonyms, and values given a corpus of documents and sample schemas [11, 37].

Data cleaning: Existing solutions include purely computational systems [31], as well as human-in-the-loop systems like DataWrangler (now Trifacta) [23] and others [20, 36]. These systems provide significant benefit, and are able to achieve high accuracy. Thus they provide us a solid foundation to clean data from data swamps. Furthermore, Bergman et al. [6] as well as Krishnan et al. [26] investigated query-based formulations of data cleaning in systems that support queries. However, since many data formats do not support queries, their techniques are not suitable for our purpose.

Data integration: There is significant support in literature on integrating data from multiple sources. Prior research has proposed two broad approaches—(i) exploiting large-scale data [5, 12] and (ii) utilizing human-in-the-loop assistance [19, 28, 34, 35]. Generally speaking, the data-driven approaches compare the similarity of individual records using a preferred metric (e.g., edit distance, Jaccard similarity), potentially assigning confidence scores and/or comparing other tuples in the record, and then deciding whether the entities match. The human-in-the-loop approaches focus more on gaining confidence on whether entities match based on data collected from crowd workers, while adjusting the model based on the worker reliability. We leverage these approaches in our work.

Data management: In our context of data swamps, the data management tasks include users keeping files [10, 22], finding files [8, 15], maintaining their file collections [32], and versioning files [7]. Notably, these tasks are closely related to the field of personal information management (PIM) [21]. Existing research on PIM systems aimed to help users better store, organize and retrieve their data in contexts like email, local files or even cloud storage [2, 4, 24]. Our work extends this research to the data swamps. Specifically our work points out concrete directions and research challenges to effectively incorporate human feedback for better data management in the data swamps.

Lastly, aside from these different components of processing data, another interesting relevant challenge is to minimize the human interactions using active learning [13, 27]. This direction is complementary to our research and advances in that field can be directly incorporated in our proposal.

7 FUTURE WORK

Based on these initial results, our next step is to perform studies of different file systems and data repositories by surveying users and requesting their feedback along the dimensions specified in the conceptual model. We intend to use these survey results to analyze the correlation between user responses and file metadata, file contents metrics, and similarity measures to investigate if human input could then lay the basis for efficient and effective methods of data discovery and data management. Implementing these solutions would then flesh out the pipeline fully enough that a data swamp navigator tool would become feasible. This, we believe, is a strong motivation and ultimate goal for our work.

8 CONCLUSION

In this paper, we have empirically demonstrated the difficulty of applying pure computational methods for similarity-based data discovery and data management. Additionally, we proposed a high-level framework for human-in-the-loop approaches to these tasks, identifying the criteria that should provide the most impact in these methods. We demonstrated a basis for this methodology, and noted suggestions for future work in conducting an online user study and developing tools for these tasks. While there is still much progress to be made in this area, we believe this work can provide a stepping stone for future results.

REFERENCES

[1] Alexandr Andoni and Piotr Indyk. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 459–468.

[2] Taiwo Ayodele, Galyna Akmayeva, and Charles A. Shoniregun. 2012. Machine learning approach towards email management. In *World Congress on Internet Security (WorldCIS-2012)*. 106–109.

[3] Sreeram Balakrishnan, Alon Y Halevy, Boulos Harb, Hongrae Lee, Jayant Madhavan, Afshin Rostamizadeh, Warren Shen, Kenneth Wilder, Fei Wu, and Cong Yu. 2015. Applying WebTables in Practice.. In *CIDR*.

[4] Deborah K. Barreau. 1995. Context As a Factor in Personal Information Management Systems. *J. Am. Soc. Inf. Sci.* 46, 5 (1995), 327–339.

[5] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *The VLDB JournalâĂŤThe International Journal on Very Large Data Bases* 18, 1 (2009), 255–276.

[6] Moria Bergman, Tova Milo, Slava Novgorodov, and Wang-Chiew Tan. 2015. Query-oriented data cleaning with oracles. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 1199–1214.

[7] Anant Bhardwaj, Amol Deshpande, Aaron J Elmore, David Karger, Sam Madden, Aditya Parameswaran, Harihar Subramanyam, Eugene Wu, and Rebecca Zhang. 2015. Collaborative data analytics with DataHub. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1916–1919.

[8] Richard Boardman and M Angela Sasse. 2004. Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 583–590.

[9] Andrei Z Broder. 2000. Identifying and filtering near-duplicate documents. In *Annual Symposium on Combinatorial Pattern Matching*. Springer, 1–10.

[10] Harry Bruce. 2005. Personal, Anticipated Information Need. *Information Research: An International Electronic Journal* 10, 3 (2005), n3.

[11] Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment* 1, 1 (2008), 538–549.

[12] Kaushik Chakrabarti, Surajit Chaudhuri, Zhimin Chen, Kris Ganjam, Yeye He, and WA Redmond. 2016. Data services leveraging Bing's data assets. *IEEE Data Eng. Bull.* 39, 3 (2016), 15–28.

[13] Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu. 2015. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In *Conference on Learning Theory*. 503–522.

[14] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. 2017. The Data Civilizer System.. In *CIDR*.

[15] Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. 2016. Stuff I've seen: a system for personal information retrieval and re-use. In *ACM SIGIR Forum*, Vol. 49. ACM, 28–35.

[16] Rihan Hai, Sandra Geisler, and Christoph Quix. 2016. Constance: An intelligent data lake system. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2097–2100.

[17] Alon Halevy, Flip Korn, Natalya F Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing google's datasets. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 795–806.

[18] Joseph M Hellerstein, Vikram Sreekanti, Joseph E Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, et al. 2017. Ground: A Data Context Service.. In *CIDR*.

[19] Shawn R Jeffery, Michael J Franklin, and Alon Y Halevy. 2008. Pay-as-you-go user feedback for dataspace systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 847–860.

[20] Shawn R Jeffery, Liwen Sun, Matt DeLand, Nick Pendar, Rick Barber, and Andrew Galdi. 2013. Arnold: Declarative Crowd-Machine Data Integration.. In *CIDR*.

[21] William Jones. 2007. Personal information management. *Annual review of information science and technology* 41, 1 (2007), 453–504.

[22] William Jones. 2010. *Keeping found things found: The study and practice of personal information management*. Morgan Kaufmann.

[23] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3363–3372.

[24] Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. 2018. Forgotten But Not Gone: Identifying the Need for Longitudinal Data Management in Cloud Storage. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM.

[25] Peter Klemperer, Yuan Liang, Michelle Mazurek, Manya Sleeper, Blase Ur, Lujo Bauer, Lorrie Faith Cranor, Nitin Gupta, and Michael Reiter. 2012. Tag, You Can See It!: Using Tags for Access Control in Photo Sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 377–386.

[26] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. 2016. ActiveClean: interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment* 9, 12 (2016), 948–959.

[27] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[28] Guoliang Li. 2017. Human-in-the-loop data integration. *Proceedings of the VLDB Endowment* 10, 12 (2017), 2006–2017.

[29] Jayant Madhavan, Shawn R Jeffery, Shirley Cohen, Xin Dong, David Ko, Cong Yu, and Alon Halevy. 2007. Web-scale data integration: You can only afford to pay as you go. CIDR.

[30] Michelle L Mazurek, Yuan Liang, William Melicher, Manya Sleeper, Lujo Bauer, Gregory R Ganger, Nitin Gupta, and Michael K Reiter. 2014. Toward strong, usable access control for shared distributed data. In *Proceedings of the 12th USENIX conference on File and Storage Technologies*. USENIX Association, 89–103.

[31] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. 2017. Holoclean: Holistic data repairs with probabilistic inference. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1190–1201.

[32] Leo Sauermann, Gunnar Aastrand Grimnes, Malte Kiesel, Christiaan Fluit, Heiko Maus, Dominik Heim, Danish Nadeem, Benjamin Horak, and Andreas Dengel. 2006. Semantic Desktop 2.0: The Gnowsis Experience. In *Proceedings of the 5th International Conference on The Semantic Web*. 887–900.

[33] Burr Settles. [n. d.]. Active Learning Literature Survey. 2010. *Computer Sciences Technical Report* 1648 ([n. d.]).

[34] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5, 11 (2012), 1483–1494.

[35] Steven Euijong Whang, Peter Lofgren, and Hector Garcia-Molina. 2013. Question selection for crowd entity resolution. *Proceedings of the VLDB Endowment* 6, 6 (2013), 349–360.

[36] Mohamed Yakout, Ahmed K Elmagarmid, Jennifer Neville, Mourad Ouzzani, and Ihab F Ilyas. 2011. Guided data repair. *Proceedings of the VLDB Endowment* 4, 5 (2011), 279–289.

[37] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. 2012. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*. ACM, 97–108.