SIGMOD Officers, Committees, and Awardees

Chair
Divyakant Agrawal
Department of Computer Science
UC Santa Barbara
Santa Barbara, California
USA
+1 805 893 4385
agrawal <at> cs.ucsb.edu

Vice-Chair
Fatma Ozcan
Systems Research Group
Google
Sunnyvale, California
USA
+1 669 264 9238
Fozcan <at> google.com

Secretary/Treasurer
Rachel Pottinger
Department of Computer Science
University of British Columbia
Vancouver
Canada
+1 604 822 0436
Rap <at> cs.ubc.ca

SIGMOD Executive Committee:

Divyakant Agrawal (Chair), Fatma Ozcan (Vice-chair), Rachel Pottinger (Treasurer), Juliana Freire (Previous SIGMOD Chair), K. Selçuk Candan (SIGMOD Conference Coordinator), Rada Chirkova (SIGMOD Record Editor), Chris Jermaine (ACM TODS Editor in Chief), Divesh Srivastava (2021 SIGMOD PC co-chair), Stratos Idreos (2021 SIGMOD PC co-chair), Leonid Libkin (Chair of PODS), Sihem Amer-Yahia (SIGMOD Diversity and Inclusion Coordinator), Curtis Dyreson (Information Director)

Advisory Board:

Yannis Ioannidis (Chair), Phil Bernstein, Surajit Chaudhuri, Rakesh Agrawal, Joe Hellerstein, Mike Franklin, Laura Haas, Renee Miller, John Wilkes, Chris Olsten, AnHai Doan, Tamer Özsu, Gerhard Weikum, Stefano Ceri, Beng Chin Ooi, Timos Sellis, Sunita Sarawagi, Stratos Idreos, and Tim Kraska

SIGMOD Information Director:

Curtis Dyreson, Utah State University

Associate Information Directors:

Huiping Cao (SIGMOD Record), Georgia Koutrika (Blogging), Wim Martens (PODS), and Sourav S Bhowmick (SIGMOD Record)

SIGMOD Record Editor-in-Chief:

Rada Chirkova, NC State University

SIGMOD Record Associate Editors:

Lyublena Antova, Marcelo Arenas, Manos Athanassoulis, Renata Borovica-Gajic, Vanessa Braganholo, Susan Davidson, Aaron J. Elmore, Wook-Shin Han, Wim Martens, Kyriakos Mouratidis, Dan Olteanu, Tamer Özsu, Kenneth Ross, Pınar Tözün, Immanuel Trummer, Yannis Velegrakis, Marianne Winslett, and Jun Yang

SIGMOD Conference Coordinator:

K. Selçuk Candan, Arizona State University

PODS Executive Committee:

Leonid Libkin (chair), Christoph Koch, Reinhard Pichler, Dan Suciu, Yufei Tao, Jan Van den Bussche

Sister Society Liaisons:

Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment), Christian Jensen (IEEE TKDE)

SIGMOD Awards Committee:

Sunita Sarawagi (Chair), Volker Markl, Renée Miller, H. V. Jagadish, Yanlei Diao, Stefano Ceri

Jim Gray Doctoral Dissertation Award Committee:

Vanessa Braganholo (co-chair), Viktor Leis (co-chair), Bailu Ding, Immanuel Trummer, Joy Arulraj, Jose Faleiro, Gustavo Alonso, Wolfgang Lehner

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Recipients of the award are the following:

Michael Stonebraker (1992)
David DeWitt (1995)
Serge Abiteboul (1998)
Rudolf Bayer (2001)
Ronald Fagin (2004)
Jennifer Widom (2007)
Umeshwar Dayal (2010)
Stefano Ceri (2013)
Gerhard Weikum (2016)
Anastasia Ailamaki (2019)
Jim Gray (1993)
C. Mohan (1996)
Hector Garcia-Molina (1999)
Patricia Selinger (2002)
Michael Carey (2005)
Moshe Y. Vardi (2008)
Surajit Chaudhuri (2011)
Martin Kersten (2014)
Goetz Graefe (2017)
Beng Chin Ooi (2020)
Philip Bernstein (1994)
David Maier (1997)
Rakesh Agrawal (2000)
Don Chamberlin (2003)
Jeffrey D. Ullman (2006)
Masaru Kitsuregawa (2009)
Bruce Lindsay (2012)
Laura Haas (2015)
Raghu Ramakrishnan (2018)

SIGMOD Systems Award

For technical contributions that have had significant impact on the theory or practice of large-scale data management systems.

Michael Stonebraker and Lawrence Rowe (2015); Martin Kersten (2016); Richard Hipp (2017); Jeff Hammerbacher, Ashish Thusoo, Joydeep Sen Sarma; Christopher Olston, Benjamin Reed, and Utkarsh Srivastava (2018); Xiaofeng Bao, Charlie Bell, Murali Brahmadesam, James Corey, Neal Fachan, Raju Gulabani, Anurag Gupta, Kamal Gupta, James Hamilton, Andy Jassy, Tengiz Kharatishvili, Sailesh Krishnamurthy, Yan Leshinsky, Lon Lundgren, Pradeep Madhavarapu, Sandor Maurice, Grant McAlister, Sam McKelvie, Raman Mittal, Debanjan Saha, Swami Sivasubramanian, Stefano Stefani, and Alex Verbitski (2019); Don Anderson, Keith Bostic, Alan Bram, Grg Burd, Michael Cahill, Ron Cohen, Alex Gorrod, George Feinberg, Mark Hayes, Charles Lamb, Linda Lee, Susan LoVerso, John Merrells, Mike Olson, Carol Sandstrom, Steve Sarette, David Schacter, David Segleau, Mario Seltzer, and Mike Ubell (2020)

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)
Jeffrey Ullman (1996)
Raghu Ramakrishnan (1999)
Daniel Rosenkrantz (2001)
Surajit Chaudhuri (2004)
Hans-Jörg Schek (2007)
David Lomet (2010)
H.V. Jagadish (2013)
Samuel Madden (2016)
Ahmed Elmagarmid (2019)
Stratos Idreos (2020)
Dennis Shasha (2020)
Gio Wiederhold (1995)
Avi Silberschatz (1997)
Michael Carey (2000)
Richard Snodgrass (2002)
Hongjun Lu (2005)
Klaus R. Dittrich (2008)
Gerhard Weikum (2011)
Kyu-Young Whang (2014)
Yannis E. Ioannidis (2017)
Philipe Bonnet (2020)
Stefan Manegold (2020)
Yahiko Kambayashi (1995)
Won Kim (1998)
Laura Haas (2000)
Michael Ley (2003)
Tamer Özsu (2006)
Beng Chin Ooi (2009)
Marianne Winslett (2012)
Curtis Dyreson (2015)
Z. Meral Özsoyoğlu (2018)
Juliana Freire (2020)
Ioana Manolescu (2020)

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **2006** *Winner*: Gerome Miklau. *Honorable Mentions*: Marcelo Arenas and Yanlei Diao
- **2007** *Winner*: Boon Thau Loo. *Honorable Mentions*: Xifeng Yan and Martin Theobald

- **2008** *Winner*: Ariel Fuxman. *Honorable Mentions*: Cong Yu and Nilesh Dalvi
- **2009** *Winner*: Daniel Abadi. *Honorable Mentions*: Bee-Chung Chen and Ashwin Machanavajjhala
- **2010** *Winner:* Christopher Ré. *Honorable Mentions*: Soumyadeb Mitra and Fabian Suchanek
- **2011** *Winner*: Stratos Idreos. *Honorable Mentions*: Todd Green and Karl Schnaitterz
- **2012** *Winner*: Ryan Johnson. *Honorable Mention*: Bogdan Alexe
- **2013** *Winner*: Sudipto Das, *Honorable Mention*: Herodotos Herodotou and Wenchao Zhou
- **2014** *Winners*: Aditya Parameswaran and Andy Pavlo.
- **2015** *Winner*: Alexander Thomson. *Honorable Mentions*: Marina Drosou and Karthik Ramachandra
- **2016** *Winner*: Paris Koutris. *Honorable Mentions*: Pinar Tozun and Alvin Cheung
- **2017** *Winne*r: Peter Bailis. *Honorable Mention*: Immanuel Trummer
- **2018** *Winne*r: Viktor Leis. *Honorable Mention*: Luis Galárraga and Yongjoo Park
- **2019** *Winne*r: Joy Arulraj. *Honorable Mention*: Bas Ketsman
- **2020** *Winner:* Jose Faleiro. *Honorable Mention:* Silu Huang
- **2021** *Winner:* Huanchen Zhang, *Honorable Mentions:* Erfan Zamanian, Maximilian Schleich, and Natacha Crooks

A complete list of all SIGMOD Awards is available at: **https://sigmod.org/sigmod-awards/**

[Last updated: September 30, 2021]

Editor's Notes

Welcome to the December 2021 issue of the ACM SIGMOD Record!

This issue starts with the Database Principles column featuring an article by Cormode on current trends in data summaries. The focus of data summarization is on finding small data structures, such as samples or sketches, that compactly represent large data sets, can be updated flexibly, and can answer accurately certain queries on the original data. Key application areas include approximate query processing, as well as distributed and stream processing. The article draws an *approximate summary* of efforts in this area, surveying the topics of summaries for machine learning, machine learning for summaries, summaries in privacy, robust streaming, and approximate counting. The author also discusses new bounds and new applications for existing summaries, and outlines open problems for future research.

The Vision column features two articles. The first article, by Mansour, Srinivas, and Hose, aims to address major open problems of handling artifacts in Open Data Science, focusing on which artifacts should be combined to achieve user goals and on how to find artifacts that are semantically similar or connected. The authors propose a federated data-science platform called KEK that closes the gap, by enabling automatic artifact location and sharing, thus breaking down silos in data science. The article details the platform and lists open research challenges and opportunities in this space.

The second article in the Vision column, by Amer-Yahia and colleagues, introduces an end-to-end data-exploration system that is able to guide users in the exploration process, by being reactive and anticipative both for data discovery and data linking. Systems with such capabilities have the potential to open productive data exploration to users with different domain and data-science expertise and experiences in various scientific communities. The system described in the article, called INODE, leverages both machine learning and semantics for data management, encapsulating domain semantics in exploration by example, by natural language, and by recommendation. The authors describe the INODE architecture and discuss challenges and opportunities that arise from the project.

The Research Articles column features an article by Bonifati, Mior, Naumann, and Noack that provides an analysis of participation of women in papers at various top-level database conferences and journals. The preliminary findings of the authors show that there is an overall growth of the number of accepted papers authored by women in major database conferences. The study also examines how the data-management field stands with respect to the fields of HCI, AI, Algorithms, Networking, and Operating Systems. The entire analysis presented in the article is reproducible, with the code and additional results being publicly available.

The Advice to Mid-Career Researchers column presents an article by Balazinska, who invites the readers promoted to senior roles to pause and reflect on where they are and what the next steps are. The issues discussed in the article include hard work in doing great things, opportunities to grow in research and to expand the types of impact that one can have, growth in mentoring of more junior researchers, and leadership in the community. The article also contains advice on overcoming stress.

The article by Arulraj published in the DBrainstorming column discusses opportunities and challenges in effective and efficient automatic video analytics. The author focuses on difficulties in processing video-analytics queries whose aim is to detect actions, as well as in training filters for each unique combination of parameters in video DBMS. The article outlines ideas for overcoming these

challenges and calls for synergistic solutions by the database, computer-vision, and machine-learning communities.

The Distinguished Profiles column features two articles. The first interview is with Juliana Freire, ACM Fellow and professor at New York University. Juliana has a Google Faculty Research Award, an IBM Faculty Award, and an NSF Career Award. She has served as chair of SIGMOD until earier this year; her Ph.D. is from Stony Brook. Juliana begins the interview by talking about her experience of battling against outdated traditions in the database-research community. She shares her thoughts on why reproducibility is needed in the fast-changing area of computer science, lists reproducibility barriers, and speaks to the imperatives of keeping research artifacts working and ensuring access to their provenance. Juliana touches on the practical impact of her VisTrails system, talks about the ReproZip project, and outlines her vision of the future of tool-based reproducibility. She then discusses the steps she has taken toward the goal of letting all flowers bloom in the community research, and shares advice for fledgling and mid-career database researchers.

The second interview in the column is with Huanchen Zhang, the 2021 winner of the ACM SIGMOD Jim Gray Dissertation award. After a postdoc at Snowflake, Huanchen is an assistant professor at Tsinghua University; his Ph.D. is from Carnegie Mellon University. In the interview, Huanchen talks about his thesis entitled Memory-Efficient Search Trees for Database Management Systems, including the three-step recipe for designing new search-tree data structures that are compact in size and, at the same time, very fast. He discusses the impact of his thesis work in industry, and provides advice for today's graduate students.

The Reports column features an article by Bonnet, Dong, Naumann, and Tözün about the experience of designing and organizing VLDB 2021 as a hybrid conference. VLDB 2021 took place in August 2021, with 180 in-person attendees in Copenhagen, Denmark and with 840 remote attendees. The article describes the key decisions of the general chairs and PC chairs of the conference, and shares the lessons learned. The authors believe that the hybrid format for scientific conferences is here to stay and opens up new opportunities for everyone.

On behalf of the SIGMOD Record Editorial board, I hope that you enjoy reading the December 2021 issue of the SIGMOD Record!

Your submissions to the SIGMOD Record are welcome via the submission site:
https://mc.manuscriptcentral.com/sigmodrecord

Prior to submission, please read the Editorial Policy on the SIGMOD Record's website:
https://sigmodrecord.org/sigmod-record-editorial-policy/

Rada Chirkova

December 2021

Past SIGMOD Record Editors:

Yanlei Diao (2014-2019)	Ioana Manolescu (2009-2013)	Alexandros Labrinidis (2007–2009)
Mario Nascimento (2005–2007)	Ling Liu (2000–2004)	Michael Franklin (1996–2000)
Jennifer Widom (1995–1996)	Arie Segev (1989–1995)	Margaret H. Dunham (1986–1988)
Jon D. Clark (1984–1985)	Thomas J. Cook (1981–1983)	Douglas S. Kerr (1976-1978)
Randall Rustin (1974-1975)	Daniel O'Connell (1971–1973)	Harrison R. Morse (1969)

Current Trends in Data Summaries

Graham Cormode*
Meta AI

ABSTRACT

The research area of data summarization seeks to find small data structures that can be updated flexibly, and answer certain queries on the input accurately. Summaries are widely used across the area of data management, and are studied from both theoretical and practical perspectives. They are the subject of ongoing research to improve their performance and broaden their applicability. In this column, recent developments in data summarization are surveyed, with the intent of inspiring further advances.

1. INTRODUCTION

The data management community makes extensive use of various kinds of summaries: compact data structures that represent a large dataset, and allow queries to be answered with some guarantee of accuracy. The most common example of summaries come in the form of samples, where evaluating a query on a sample provides an approximate answer to the query on the full data set. Other popular summary types are Bloom filters [8], which approximately represent sets, and sketches [12], which approximately represent vectors, as well as other summaries targeting more specific queries. Key application areas include approximate query processing (AQP), where sampling is quite ubiquitous [43], and distributed and stream processing [25].

The design and application of summaries is now ubiquitous within the research community, and has been the subject of several tutorials and books, covering developments from the late 1970s onwards [60, 45, 56, 19]. In this column, I will give a very high-level survey of current active research directions in data summarization, with emphasis on results from the last few years. This is a very subjective and partial view, based on topics that have been the focus of recent papers in data management venues, or just ones that have caught the interest of researchers in this area. The intent is, fittingly, to draw an approximate summary of efforts in this area, rather than a precise characterization.

*gcormode@fb.com

2. SUMMARIES FOR ML

Given the high level of interest in machine learning (ML) across computer science and beyond, it should be no surprise that researchers are looking to use data summaries in order to improve the ML training process. The primary application of summaries is to try to reduce the size of ML models without sacrificing their expressivity. The most natural place to apply data summaries is in compressing the information exchanged between data owners during the training of networks. In distributed training of machine learning models (usually referred to as Federated Learning [37]), each client holds some labeled examples, and a server sends out a candidate model. Each client evaluates the candidate model on their labeled examples, and determines an update to the model, typically in the form of a gradient vector to adjust the model parameters in order to improve the accuracy of the model on their examples. The server will then update the model based on combining these gradients, often by moving in the direction of the average gradient. However, the size of the model can be very large, and sending the full gradient vector can have high computational cost for each client (in terms of uplink communication). It is natural to look to data summaries as a way to reduce the size of the communication, with the tradeoff of potentially slightly increasing the number of steps before the model converges, or of slightly reducing the accuracy of the final model that is found.

Two recent papers suggest similar approaches to reducing communication in Federated Learning with the use of sketches. In FetchSGD [52], the authors propose the use of the CountSketch summary [12] as the medium through which to convey the gradient updates. CountSketch has several attractive features: it promises to preserve the large entries of the input vector accurately, and so using sketches captures the most significant parts of the updates. In addition, it is a linear summary: sketches can be summed and subtracted, with the resulting sketch being identical to the one we would obtain if we had applied these operations to the input vectors before sketching. This means that we can treat the

sketches as if they were the full vectors, and apply various techniques from machine learning, such as momentum (including updates from previous iterations at lower weight) and error compensation. In addition, it is possible to prove results on the speed and accuracy of convergence under standard ML assumptions.

The FedSketch [28] paper follows a similar outline, also making use of CountSketch as a compression operator. It additionally considers the provision of a differential privacy guarantee, taking advantage of both linearity and the sparsity of the CountSketch transformation. Experiments and analysis demonstrate that this approach converges more quickly than other previously proposed private federated approaches. Away from the federated setting, Tai *et al.* propose the Weight-Median sketch as a tool for sketching gradients, which is applied to learn linear classifiers over streams of updates [54].

There are a number of other directions in which summarization can assist in machine learning. An orthogonal approach to handling the large size of ML models in the literature is to apply quantization to the model parameters. That is, rather than representing each parameter with a 32 or 64 bit floating point representation, they can instead be represented more crudely by a much fewer number of bits. Currently popular approaches apply fairly simple quantization encoding – for example, using 8 bits to represent values divided uniformly between a minimum and maximum value. This approach is rather coarse, and can lead to errors accumulating when multiple quantized update vectors are combined together. A more promising approach might be to use randomized representations of values, so that errors tend to cancel out on average as more vectors are aggregated [58]. Similarly, pruning is a simple way to reduce the size of an update vector. Under pruning, values in update vectors with small magnitude are pruned to zero, and can be omitted from reporting back to the server. An intriguing open research direction would be to combine pruning with techniques from data summarization (e.g., sketching), to more compactly encode the sparse pruned updates.

3. ML FOR SUMMARIES

Just as summarization can help with machine learning, so too can machine learning help summarization. A highly impactful paper from 2018 argued that rather than traditional indices (B-trees and the like), it is valuable to use compact models to access data [41]. That is, train a model to predict where to find a piece of data, by minimizing an appropriate loss function, since all indices can be interpreted as implicit models of the data layout. One way to "train" a Bloom filter is to optimize the hash functions: to define a hash function via a machine learning model (a neural network), which is optimized to reduce the number of false positives for a given set of data.

This notion has been generalized to a wider range of summaries. Hsu *et al.* considered sketches for frequencies [30]. Similar to the Bloom filter case, the aim is to choose a hash function that gives better results for a data distribution than choosing a random hash function. The authors show that it is indeed possible to "learn" a good hash function, and analyze the resulting error under some assumptions on this distribution. Jiang *et al.* [35] expanded the applicability of this approach to a range of other summary types, such as distinct counting and frequency moments. In more detail, the approach is to assume the existence of a "frequency oracle" for the distribution, so that given an item the oracle accurately predicts the frequency of this item in the full distribution. By handling items differently based on their predicted frequency, it is possible to obtain bounds on the size of summaries better than those in the general case without such an oracle.

This paradigm has sparked work in other directions, notably for linear algebra involving large matrices. Indyk *et al.* [33] consider learning a low-rank approximation of a matrix, aiming to minimize the Frobenius norm of the difference between the original and approximate matrix. The approach is to learn a sketch projection matrix through which to generate the approximation. It is observed that the error can be reduced by up to an order of magnitude compared to a randomly chosen sketch. Li *et al.* [44] similarly consider sketches for the Hessian of matrices, and apply these to ML problems such as regularized regression (LASSO) and matrix regression. ML techniques have even been applied to learn how to multiply matrices (Blalock and Guttag [7]): here, the aim is to learn functions that can be applied to matrices A and B so as to allow a fast construction of a matrix C that is close to AB under the Frobenius norm.

It will be interesting to think more generally about summaries augmented with an oracle that (accurately or perfectly) captures some part of the problem being studied, to understand the impact of the hardness of the task. This can be viewed as a different kind of assumption compared to promises on the arrival order of data items (arbitrary, random or worst-case) or on the statistical distribution of data values that have been made in prior work (e.g., [27, 17]). Graphs and matrices are natural candidates: how well can we summarize the structures if we have, for example, a shortest path oracle, or access to the eigenvalues?

4. SUMMARIES IN PRIVACY

The objective of privacy enhancing technologies is to limit the amount of information revealed to an

observer, while the objective of data summarization is to support answering a particular query while limiting the amount of information retained. There is sufficient alignment from these two objectives that it is feasible to use data summaries as part of a privacy solution to assist with the information limitation. This has led to a number of advances in privacy technology. The large scale deployments of private data collection by Google [21] and Apple [2], which both relied on the use of summaries, meant that these were some of the most high-profile applications for data summaries. Specifically, the Rappor system from Google was built on Bloom Filters [8], while the Apple implementation made use of sketches to bound the dimensionality of the data gathered [19]. These two examples were both primarily concerned with gathering frequency statistics from high dimensional distributions, to find the heavy hitters from the input via so-called "frequency oracles" in the local model of differential privacy. Bassily *et al.* formalized this approach in their analysis [4].

More generally, there has been a growth in interest in the area of Federated Analytics (FA), which seeks to gather information from multiple distributed clients in order to provide statistics on the union of their inputs. Unsurprisingly, data summaries can be employed in the construction of federated analytics protocols. The demands of FA go beyond those for summaries that can be constructed independently and merged centrally. Typically, we seek some additional guarantee of privacy. A clear example is given by the TrieHH protocol proposed by researchers at Google [62]. Here, the aim is to find the set of heavy hitter items from a large collection. The general approach is to gather information from distributed clients in order to search for heavy hitters in a hierarchical fashion, similar to approaches performed in the data streaming setting. However, the set of candidate items is identified by a sampling step, with a novel proof that those items whose frequency in the sample exceeds a threshold achieve a (centralized) differential privacy guarantee, without the need for explicit noise addition.

Recently, there has also been interest in studying the inherent privacy offered by data summaries. The intuition is clear: when summaries store very compact information about their input, it is natural to imagine that the information retained about any given input item should be quite small, and hence private. Formalizing this intuition, and ensuing that it is not possible to "invert" the summarization process to recover the input items, requires considerable care and effort. Recent results on approximate counting have shown that the Flajolet-Martin summary achieves a level of differential privacy – provided that the observer does not know which hash functions were used to create the summary (which is assumed to be a uniform random permutation), and the cardinality of items being summarized is not too small [53, 13]. This refines the work of Desfontaines *et al.* [20], which showed that applied directly, many distinct count sketches do *not* provide a privacy guarantee. Most recently, Pagh and Stausholm give a sketch for this problem with privacy guarantees where the hash function can be known to the adversary, and privacy is achieved by perturbing the stored information, i.e., by applying randomized response to the stored bits [49]. This enables private sketches to be shared between multiple parties in order to approximate the cardinality of unions of sets.

Two other foundational summarization tasks are sampling and counting. Work by Cohen *et al.* [14] looks at private sampling from weighted inputs, where the weights can be thought of as the number of individuals who hold a particular item. The aim is to produce a compact collection of items and noisy weights, so that the collection functions as a good sample of the input (representing the weight distribution), while protecting the privacy of individuals who contributed the data. This means that particular care has to be taken to ensure that low weights do not reveal information about the data of the participants. The essence of the approach is to define inclusion probabilities for elements based on weights which achieve both sampling accuracy and differential privacy. In particular, a sampling scheme is defined such that sampling probabilities for weights that differ by one meet the (approximate) differential privacy definition. The approach inherits many of the benefits of (non-private) sampling, such as accurate estimators for linear statistics, and gives solution for many private tasks, such as quantiles and histograms.

Gathering accurate (private) statistics in the distributed setting while minimizing communication naturally benefits from data summarization techniques. This gives the multiparty differential privacy model, which generalizes both the local model (where each of k users holds a single item) and the central model (where multiple items are held by a single entity). Recent work makes use of the Count Sketch, whose sparsity means that it has low sensitivity under differential privacy [31]. Instead of merging the sketches as in a standard linear sketch by using the same set of parameters (sketch size and hash functions), the construction uses different parameters for each user based on the size of their input, and combines the estimates from each sketch with an additional error bound. This approach saves a $\sqrt{k}$ factor in the multiparty model, and achieves an optimal error-communication tradeoff.

It is natural to ask what other problems with a privacy requirement can be helped by the use of summaries, or other ideas inspired by summarization. A particular challenge in privacy is handling

longitudinal data, i.e., situations where a user participates in the data collection multiple times as time goes on, but we wish to give an overall guarantee on the privacy despite a potentially unbounded influence on the data. There have certainly been efforts to address this concern, but the approaches deployed in practice are not entirely satisfying, relying either on "resetting" the privacy budget on a daily basis, or using a somewhat heuristic memoization of random values [2, 21]. The basic idea of keeping a tree-structure over continually observed to reduce the noise to logarithmic [11] has been widely used for similar purposes, most recently in the context of federated learning [36].

5. NEW MODELS: ROBUST STREAMING

One of the core areas that motivates the development of new summary structures is the area of data stream processing. Here, the aim is to summarize a large input arriving as a stream of inputs, in order to answer a basic query, such as estimating the frequency moments of the data distribution. Traditionally, summaries have been analyzed assuming that the stream may be arbitrary, but is fixed independent of the random choices of the summarization algorithm. This allows effective randomized algorithms to be proposed with strong space-accuracy tradeoffs. However, there are cases where this may seem overly optimistic: when the data structure is queried during the arrival of the stream, knowledge of the approximate answer could be used to influence the subsequent items in the input, and elicit an erroneous answer. To ensure the highest level of reliability, we might ask whether it is possible to design summary techniques that are robust to inputs that are chosen adversarially, in reaction to the actions of the algorithm. A starting point is deterministic algorithms: any approach which gives a guarantee that holds over all possible inputs is necessarily robust to adversarial inputs. However, for many fundamental problems in streaming, it is known that there is a large gap between deterministic and randomized bounds, where often no deterministic algorithm can do better than storing the whole input.

A recent line of work has considered this question, and shown that it is possible to construct summaries that are indeed robust in this fashion, with a moderate overhead compared to their non-robust alternatives. Ben-Eliezer and Yogev [6] first considered the adversarial robustness of sampling. It is perhaps not very surprising that drawing a random sample of a stream of data is fairly robust to an adversary choosing the input items, since the sampling is performed without close inspection of any item. However, one could envision an adversary who observes the current state of the sample, and chooses input items in order to try to exaggerate any ways in which the sample is already misrepresentative. The results of Ben-Eliezer and Yogev prove that nevertheless, to evade any such adversary, the sample only needs to be a small factor larger than in the non-adversarial case.

A subsequent work of Ben-Eliezer *et al.* [5] considers a broader range of problems, such as frequency moments, distinct counting and frequency estimation, in the adversarial setting. This work was recognized as the best paper of PODS 2020. The central result is a generic framework which introduces the parameter of the *flip number*. This counts how often the answer of the algorithm must change over the course of observing its input. Since we typically consider approximate algorithms, it is often the case that the summary can give the same output for an extended period while still meeting the required approximation bounds. Consider, for example, the (trivial) streaming algorithm to count the number of items observed so far, n. We can observe that to give a 2-factor approximation, the flip number can be bounded to $O(\log n)$ (we only have to change the output after the input size has doubled). More sophisticated arguments serve to bound the flip number for more challenging functions. The paper then argues that it suffices to run multiple copies of a (non-adversarially robust) streaming summary in parallel. We can report the output of one summary while it is an accurate enough approximation of the true answer, then switch to a 'fresh' instance when this changes. The number of summaries to maintain is then linear in the flip number of the problem considered.

Subsequent work has built on this foundation. Hassidim *et al.* make an intriguing connection between robustness and privacy, by employing differential privacy to thwart the adversary [29]. Specifically, the technique also runs multiple copies of non-adversarial streaming algorithms for the problem, but then aggregates their output in a way that provides a differential privacy guarantee. The intent is that the adversary, observing the changing output of the algorithm, is nevertheless unable to draw strong inferences about the inner state of the various summaries due to the privacy noise. Significantly, the cost of the approach also depends on the flip number, but is now proportional to the square root of the flip number. Another surprising connection work that draws a link between adversarial sampling and the theory of online learning [1]. It shows that the concepts for which there exist effective adversarially robust sampling mechanisms are those that meet a definition of online learnability. Braverman *et al.* have demonstrated that the commonly used technique of "merge and reduce" to build summaries over distributed data brings with it a guarantee of adversarial robustness, providing strong guarantees for various clustering problems such as

k-means, k-median, k-center and more [10]. Meanwhile, Woodruff and Zhou showed tighter bounds for various problems in the sliding window streaming model [59]. A strong separation was shown between the adversarial and non-adversarial model by Kaplan *et al.* [38], by considering the "adaptive data analysis" problem, which can be shown to require exponentially more space in the adversarial setting.

There are many open directions in the area of robust streaming, as evidenced by a recent workshop day dedicated to the topic[1]. Some immediate directions are to understand the true dependence on the flip number in the space bounds. Is it too much to hope for a polylogarithmic bound by keeping this many instances of independent summaries, and selecting random subsets of these to provide an estimate? More generally, could the notion of using differential privacy as a tool to fool adversaries have wider applicability?

6. PROGRESS IN APPROXIMATE COUNTING

Counting is one of the most basic computational tasks, so it is hard to imagine that there would be new progress on it. Nevertheless, in the last few years there have been some intriguing new steps made for counting, specifically on various notions of approximate counting. Approximate counting via the Morris counter is often used as an example in a randomized algorithms class [47]. The algorithm keeps a counter with a small bit depth, and processes increment updates. The internal counter is incremented with probability that decreases exponentially with its value. This can be used to estimate quantities with value up to n using bit depth of only $O(\log \log n)$. Recently, Nelson and Yu [48] revisited this problem, and showed tighter bounds on the accuracy of such counters. In particular, they showed a new algorithm with a simple proof that uses space $O(\log 1/\epsilon + \log \log 1/\delta + \log \log n)$ in order to approximate a quantity up to n with $1 \pm \epsilon$ accuracy with probability $1 - \delta$. They go on to show via a more involved proof that the same bound holds for a lightly modified version of the original Morris algorithm. This improves the dependency on δ exponentially. Offering accurate approximate counters in small space is of value to data science applications which maintain a large number of counters for many different events in parallel.

In a different setting, recent work has tried to reduce the size of counters down to a single bit. Specifically, we have a number of participants who each hold a real value x, scaled to the range $[0, 1]$, and our aim is to gather information from them in order to estimate the mean of their (scaled) values. A simple randomized rounding approach is to round x to 1 with probability x, and 0 otherwise: the expectation of this rounding is x. Ben Basat *et al.* [3] consider a variety of related approaches, and show that variance of $\frac{1}{4}$ of the simple rounding approach can be improved in situations when shared randomness is available, or a biased estimator can be adopted. Note that limiting to a single random bit alone may not make a big difference to communication cost: the overheads in packet-switched networks are such that the difference between sending 1 bit vs. 64 bits is small compared to the cost of packet headers etc. However, this approach offers clearer benefits when sending larger volumes of data (say, a vector of values), or when we want to apply privacy to the transmitted bits, and can randomly noise the bit that is sent.

The counting problem becomes more challenging when we have to address the problem of distinct counting: given an unsorted collection of items (with some repeated), we seek to estimate the cardinality of the support set. This problem appears in many applications where summaries are desirable, and many effective algorithms have been proposed. Perhaps the most famous of these is the HyperLogLog summary presented by Flajolet et al. [23]. A recent advance on this problem is due to Pettie and Wang, who seek to understand tight bounds for the space complexity of this problem – again, this is a pressing concern when maintaining approximate (distinct) counters for a large number of different objects [50]. In particular, they show a new approach to analyzing the space complexity by fusing the Fisher information with the Shannon entropy of the summary. This enables them to revisit the exact constants of an algorithm due to Flajolet and Martin [24], when implemented in a compressed form. Under some restrictions, they show that this sketch is optimal (including the constant factor), which settles a long line of work seeking increasingly tight bounds for this problem. Rather than being a theoretical observation about an impractical algorithm, the "compressed probabilistic counting" technique was already implemented in the Apache data sketches library[2], and has been used internally within Oath (Yahoo!) for monitoring large volumes of statistics. The analytical study of Pettie and Wang complemented the numerical study of Lang, who implemented and evaluated this algorithm [42]. In subsequent work, Pettie *et al.* went on to study the space complexity of non-mergable summaries for distinct counting, and show that sacrificing mergability can obtain slightly higher space efficiency for summaries [51].

The next step might be to move these advances in approximate counting closer to applications. As noted above, the importance of machine learning, which relies in part on large collections of numeric

[1] `https://rajeshjayaram.com/stoc-2021-robust-streaming-workshop.html`

[2] `http://datasketches.apache.org`

values, is a strong candidate to benefit from approximate counters, either during training, or after training for efficient communication and storage on devices. More generally, the proliferation of data means that it is ever easier to capture and store large volumes of data should provide an important use-case for approximate counting in various forms, particularly to handle counters which vary frequently over time. It would be particularly compelling to see empirical evidence of the benefits of using approximate counting in practice.

7. PROGRESS IN QUANTILES

Given a collection of data items from an ordered domain, the quantiles characterize the cumulative distibution function (CDF) of the empirical distribution. In simpler terms, they capture the median, and more generally the percentiles of the data. Given a fixed data set, finding the quantiles can be done easily if it is feasible to sort the data, and with more effort without sorting by a classical linear time algorithm [9]. However, in the context of summarization, we often seek a compact summary that can be created from a stream of updates, or by merging summaries of subsets of the dataset together, without having random access to the dataset in full. Until recently, the state of the art was generally considered to be the Greenwald-Khanna summary (from 2001) [26], and the KLL summary (from 2016) [39]. Both give an additive guarantee as a function of a parameter ϵ: given a target quantile, they guarantee to return an item whose rank in the sorted order of n items is at most ϵn from the target. The GK summary provides a deterministic guarantee with an $O(\frac{1}{\epsilon}\log \epsilon n)$-sized summary, while the KLL summary gives a randomized guarantee with an $O(\frac{1}{\epsilon})$-sized summary.

A number of recent advances have enhanced our understanding of this problem. From PODS 2020, a new result showed that the GK summary is essentially optimal among algorithms which only perform comparisons between items to determine what summary to retain [18]. The main result in the paper is an intricate construction based on white-box knowledge of the operation of a quantile algorithm, to construct paired inputs that maximize the error of a deterministic summary. It proceeds recursively to obtain the $\log \epsilon n$ factor in the lower bound, improving over both the trivial $\Omega(1/\epsilon)$ lower bound, and a more involved bound of $\Omega(\frac{1}{\epsilon}\log(1/\epsilon))$ that is nevertheless independent of the input size [32]. The deterministic lower bound can also be applied in the very low failure probability regime, to provide a lower bound for randomized algorithms, and so shows that the KLL summary is similarly optimal when the error probability is exponentially small.

Other advances on quantiles have considered variations of the problem and showed new results by adapting the KLL algorithm. Zhao *et al.* [61] propose "KLL ±", which accepts an input consisting of a mixture of insertions and deletions. Handling an arbitrary number of deletions can be hard: consider an input which deletes all but an arbitrary handful of items. To give a quantile guarantee on this input, the algorithm must be able to retrieve exactly the set of items which survive to the end. Instead, it is more feasible to consider the case of *bounded deletions*, where the number of deletions is promised to be at most $1 - 1/\alpha$, for a parameter α. The algorithm applies a variant of the KLL algorithm to the stream of insertions and deletions, and drops tuples when an insertion, deletion pair for the same item are placed together in the data structure. The result is shown to provide the desired additive ϵ guarantee with space $\tilde{O}(\frac{\alpha^{1.5}}{\epsilon})$.

A different goal is to provide a *relative error* guarantee for quantiles. That is, instead of answering a query with an item a fixed distance from the target quantile, we seek an item whose distance is a small fraction of the true rank of the target. This is important for cases where we seek to find accurate answers for items in the tail of the distribution, i.e., the 99^{th}, 99.9^{th} and 99.99^{th} percentiles. The problem is challenging, since if we do not retain accurate enough information on items that need high precision, we cannot hope to remedy this deficit. The "relative error quantiles sketch", which adapts the structure of the KLL algorithm to provide this improved accuracy guarantee was given the best paper award in PODS 2021 [15]. The space bound achieved is $O(1/\epsilon \log^{3/2} \epsilon n)$, which improves on prior bounds of $O(1/\epsilon \log^3 \epsilon n)$, and is close to the trivial lower bound of $\Omega(1/\epsilon \log \epsilon n)$.

There are many natural questions for this line of work. Most obviously would be to understand whether the $\log^{3/2} \epsilon n$ can be reduced closer to $\log \epsilon n$, or whether this unusual exponent is inherent. It would also be desirable to streamline and simplify the construction and its proof. In particular, the argument that instances of the relative error quantiles sketch can be merged together is very intricate. This is not to say that the algorithm itself is impractical: it has been implemented within the Apache DataSketches library[3], and used within Splunk for tracking distributions to monitor for changes. A recent empirical study compared the algorithm to a popular alternative approach, the t-digest, and showed that while the t-digest does well on "typical" inputs, there are adversarially crafted inputs on which the t-digest can be made to give extremely high error, while the relative error quantiles sketch maintains the same level of accuracy throughout [16]. Consequently, it would be highly desirable to build a summary that obtains the best of both worlds: small space and high accuracy on typical inputs,

[3] `https://datasketches.apache.org/`

while retaining space and accuracy guarantees even against worst-case inputs.

8. IMPROVEMENTS WITH EXISTING SUMMARIES: NEW BOUNDS AND NEW APPLICATIONS

One reasons for the popularity of summaries in practice is that they often give accurate results even with only small amounts of space allocated. This is in part because they follow the behavior predicted by their theoretical analysis, and often the analysis is fairly tight. That is, rather than being governed by bounds in big-Oh notation with hidden constants, we often understand their costs in closed form, with quite small explicit constants. Still, there is the strong desire to further close the gap between the good performance seen in practice and the worst-case bounds from analysis, to allow even tighter provisioning of resources for the summaries (i.e., allocate the smallest space possible to achieve the desired level of accuracy).

A good example is the Count-Min sketch, a very simple randomized summary. The original analysis uses elementary tools (such as the Markov inequality) to give a strong accuracy bound on a simple biased estimator with explicit constants. More recently, Ting [55] revisited this structure and proposed new estimators for the same sketch which provide more accurate and unbiased estimators for frequency estimation. The analysis makes use of statistical tools such as the bootstrap to provide a data-dependent error guarantee. In particular, it uses information from the parts of the sketch that do not directly answer the query in order to build an improved estimator.

Other works have sought to apply similar tools from statistics in order to give improved bounds. As described in Section 6, Nelson and Yu give improved bounds for the approximate Morris counter [48]. Ertl [22] analyzed distinct counters for the task of estimating the size of intersections between sets. This is a problem with strong lower bounds, since the intersection size can be small while the sets can be large, and so obtaining relative error is not possible. While presenting a new sketch for this problem, Ertl proposed a more general closed-form estimator that can be applied to existing sketches, such as the popular HyperLogLog summary [23]. Lopes *et al.* similarly consider sketches for matrix computations such as least-squares regression, and use a bootstrap-based approach to provide error estimates for the approximate solution. A key feature is that bootstrap is used here to understand the random variation due to a randomization in the algorithm, rather than variation in the data.

Summary techniques are increasingly finding new applications in other areas to help improve bounds. A very partial sampling of these includes:

- Sketches to help solve linear programs [57], making use of the count-sketch summary [12], taking advantage of its ability to accurately capture the heavy hitters.
- Sketches for approximate pattern matching under string edit distance [40], by summarizing strings with low edit distance to the input string.
- Solving regression problems on data that is represented in a factorized format via sketching [34].
- Using sketches to understand the trade-off between distortion and communication in voting situations [46].

We conclude with some natural (if generic) open questions: for what other summary techniques can we obtain improved bounds by exploiting more advanced analysis techniques? What new applications of summaries can there be, across the important areas of optimization, string processing, and graph and linear algebra computations? A more extensive list of open questions, covering a range of topics in sublinear algorithms, can be found at `sublinear.info`.

Acknowledgements.
Thanks to Justin Thaler, Ke Yi, Daniel Ting, David Woodruff, Jelani Nelson, Vladimir Braverman, and Christian Konrad for their suggestions of papers to highlight in this column. Thanks to Dan Olteanu for suggesting this column and providing helpful feedback.

9. REFERENCES

[1] N. Alon, O. Ben-Eliezer, Y. Dagan, S. Moran, M. Naor, and E. Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 447–455. ACM, 2021.

[2] Apple Differential Privacy Team. Learning with privacy at scale. `https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf`, 2017.

[3] R. B. Basat, M. Mitzenmacher, and S. Vargaftik. How to send a real number using a single bit (and some shared randomness). In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 25:1–25:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[4] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta. Practical locally private heavy

hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42, 2020.

[5] O. Ben-Eliezer, R. Jayaram, D. P. Woodruff, and E. Yogev. A framework for adversarially robust streaming algorithms. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 63–80. ACM, 2020.

[6] O. Ben-Eliezer and E. Yogev. The adversarial robustness of sampling. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 49–62. ACM, 2020.

[7] D. W. Blalock and J. V. Guttag. Multiplying matrices without multiplying. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 992–1004. PMLR, 2021.

[8] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[9] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, August 1973.

[10] V. Braverman, A. Hasidim, Y. Matias, M. Schain, S. Silwal, and S. Zhou. Adversarial robustness of streaming algorithms through importance sampling. In *NeurIPS*, 2021.

[11] T. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011.

[12] M. Charikar, K. C. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

[13] S. G. Choi, D. Dachman-Soled, M. Kulkarni, and A. Yerukhimovich. Differentially-private multi-party sketching for large-scale statistics. *Proc. Priv. Enhancing Technol.*, 2020(3):153–174, 2020.

[14] E. Cohen, O. Geri, T. Sarlós, and U. Stemmer. Differentially private weighted sampling. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pages 2404–2412. PMLR, 2021.

[15] G. Cormode, Z. S. Karnin, E. Liberty, J. Thaler, and P. Veselý. Relative error streaming quantiles. In *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 96–108. ACM, 2021.

[16] G. Cormode, A. Mishra, J. Ross, and P. Veselý. Theory meets practice at the median: A worst case comparison of relative error quantile algorithms. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2722–2731. ACM, 2021.

[17] G. Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *Proceedings of the 2005 SIAM International Conference on Data Mining, SDM*, pages 44–55. SIAM, 2005.

[18] G. Cormode and P. Veselý. A tight lower bound for comparison-based quantile summaries. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 81–93. ACM, 2020.

[19] G. Cormode and K. Yi. *Small summaries for big data*. CUP, 2020.

[20] D. Desfontaines, A. Lochbihler, and D. A. Basin. Cardinality estimators do not preserve privacy. *Proc. Priv. Enhancing Technol.*, 2019(2):26–46, 2019.

[21] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067. ACM, 2014.

[22] O. Ertl. Setsketch: Filling the gap between minhash and hyperloglog. *Proc. VLDB Endow.*, 14(11):2244–2257, 2021.

[23] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. *Discrete Mathematics and Theoretical Computer Science Proceedings*, page 127–146, 2007.

[24] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.

[25] M. Fragkoulis, P. Carbone, V. Kalavri, and A. Katsifodimos. A survey on the evolution of stream processing systems. *CoRR*, abs/2008.00842, 2020.

[26] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 58–66. ACM, 2001.

[27] S. Guha and A. McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. Comput.*, 38(5):2044–2059, 2009.

[28] F. Haddadpour, B. Karimi, P. Li, and X. Li. Fedsketch: Communication-efficient and private federated learning via sketching. *CoRR*, abs/2008.04975, 2020.

[29] A. Hassidim, H. Kaplan, Y. Mansour, Y. Matias, and U. Stemmer. Adversarially robust streaming algorithms via differential privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020.

[30] C. Hsu, P. Indyk, D. Katabi, and A. Vakilian. Learning-based frequency estimation algorithms. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net, 2019.

[31] Z. Huang, Y. Qiu, K. Yi, and G. Cormode. Frequency estimation under multiparty differential privacy: One-shot and streaming. *CoRR*, abs/2104.01808, 2021.

[32] R. Y. S. Hung and H. Ting. An $\omega(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ space lower bound for finding ϵ-approximate quantiles in a data stream. In *Frontiers in Algorithmics, 4th International Workshop, FAW 2010,*, volume 6213 of *Lecture Notes in Computer Science*, pages 89–100. Springer, 2010.

[33] P. Indyk, A. Vakilian, and Y. Yuan. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 7400–7410, 2019.

[34] R. Jayaram, A. Samadian, D. P. Woodruff, and P. Ye. In-database regression in input sparsity time. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 4797–4806. PMLR, 2021.

[35] T. Jiang, Y. Li, H. Lin, Y. Ruan, and D. P. Woodruff. Learning-augmented data stream algorithms. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[36] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. Practical and private (deep) learning without sampling or shuffling. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 2021.

[37] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

[38] H. Kaplan, Y. Mansour, K. Nissim, and U. Stemmer. Separating adaptive streaming from oblivious streaming using the bounded storage model. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference*, volume 12827 of *Lecture Notes in Computer Science*, pages 94–121. Springer, 2021.

[39] Z. S. Karnin, K. J. Lang, and E. Liberty. Optimal quantile approximation in streams. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pages 71–78. IEEE Computer Society, 2016.

[40] T. Kociumaka, E. Porat, and T. Starikovskaya. Small space and streaming pattern matching with k edits. *CoRR*, abs/2106.06037, 2021.

[41] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018*, pages 489–504. ACM, 2018.

[42] K. J. Lang. Back to the future: an even more nearly optimal cardinality estimation algorithm. *CoRR*, abs/1708.06839, 2017.

[43] K. Li and G. Li. Approximate query processing: What is new and where to go? - A survey on approximate query processing. *Data Sci. Eng.*, 3(4):379–397, 2018.

[44] Y. Li, H. Lin, and D. P. Woodruff. Learning-augmented sketches for hessians. *CoRR*, abs/2102.12317, 2021.

[45] E. Liberty and J. Nelson. Streaming data mining. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2012.

[46] D. Mandal, N. Shah, and D. P. Woodruff. Optimal communication-distortion tradeoff in voting. In *EC '20: The 21st ACM Conference on Economics and Computation*, pages 795–813. ACM, 2020.

[47] R. H. Morris Sr. Counting large numbers of events in small registers. *Commun. ACM*, 21(10):840–842, 1978.

[48] J. Nelson and H. Yu. Optimal bounds for approximate counting. *CoRR*, abs/2010.02116, 2020.

[49] R. Pagh and N. M. Stausholm. Efficient differentially private F0 linear sketching. In *24th International Conference on Database Theory, ICDT*, volume 186 of *LIPIcs*, pages 18:1–18:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[50] S. Pettie and D. Wang. Information theoretic limits of cardinality estimation: Fisher meets shannon. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 556–569. ACM, 2021.

[51] S. Pettie, D. Wang, and L. Yin. Non-mergeable sketching for cardinality estimation. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 104:1–104:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[52] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8253–8265. PMLR, 2020.

[53] A. D. Smith, S. Song, and A. Thakurta. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. In *Annual Conference on Neural Information Processing Systems*, 2020.

[54] K. S. Tai, V. Sharan, P. Bailis, and G. Valiant. Sketching linear classifiers over data streams. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018*, pages 757–772. ACM, 2018.

[55] D. Ting. Count-min: Optimal estimation and tight error bounds using empirical error distributions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*, pages 2319–2328. ACM, 2018.

[56] D. Ting, J. Malkin, and L. Rhodes. Data sketching for real time analytics: Theory and practice. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3567–3568. ACM, 2020.

[57] J. van den Brand, Y. T. Lee, A. Sidford, and Z. Song. Solving tall dense linear programs in nearly linear time. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 775–788. ACM, 2020.

[58] S. Vargaftik, R. B. Basat, A. Portnoy, G. Mendelson, Y. Ben-Itzhak, and M. Mitzenmacher. DRIVE: one-bit distributed mean estimation. *CoRR*, abs/2105.08339, 2021.

[59] D. P. Woodruff and S. Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. *CoRR*, abs/2011.07471, 2020.

[60] K. Yi. Random sampling on big data: Techniques and applications. In *Proceedings of the ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017*, 2017.

[61] F. Zhao, S. Maiyya, R. Weiner, D. Agrawal, and A. E. Abbadi. KLL±: Approximate quantile sketches over dynamic datasets. *Proc. VLDB Endow.*, 14(7):1215–1227, 2021.

[62] W. Zhu, P. Kairouz, B. McMahan, H. Sun, and W. Li. Federated heavy hitters discovery with differential privacy. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 108, pages 3837–3847. PMLR, 2020.

Federated Data Science to Break Down Silos [Vision]

Essam Mansour
Concordia University, Canada

Kavitha Srinivas
IBM Research, USA

Katja Hose
Aalborg University, Denmark

ABSTRACT

Similar to Open Data initiatives, data science as a community has launched initiatives for sharing not only data but entire pipelines, derivatives, artifacts, etc. (Open Data Science). However, the few efforts that exist focus on the technical part on how to facilitate sharing, conversion, etc. This vision paper goes a step further and proposes KEK, an open federated data science platform that does not only allow for sharing data science pipelines and their (meta)data but also provides methods for efficient search and, in the ideal case, even allows for combining and defining pipelines across platforms in a federated manner. In doing so, KEK addresses the so far neglected challenge of actually finding artifacts that are semantically related and that can be combined to achieve a certain goal.

1. INTRODUCTION

Open Data initiatives have led to the development of Open Data portals that provide machine-readable and structured datasets in topics, such as health, education, transportation, agriculture, and food. They are driven, for example, by governments, e.g., USA [40], Canada [6], or organizations, such as WHO [45] and WTO [46], and provide access to thousands of datasets. Encouraged by the availability of this data and the FAIR principles [44], data science projects are increasingly striving at making datasets and related data science experimentation automatically and efficiently findable, accessible, interoperable, and reusable. This includes sharing data science pipelines and derived insights, such as code, notebooks, datasets, and technical papers.

Unfortunately, despite artifacts of experimentation and creation of pipelines becoming increasingly more open, most of the artifacts are scattered across various open source repositories, such as GitHub or GitLab. Furthermore, documentation describing the work is available along with code on Jupyter notebooks, blogs in domains, such as Medium, and open repositories of preprints, e.g., ArXiv. Recently, we have therefore seen the rise of initiatives and projects, such as Agora [39], with the goal of providing the foundations of how to technically combine data science pipelines in decentralized and dynamic environments, where data, algorithms, etc. are distributed. While these projects concentrate on the question *how to technically combine artifacts*, they neglect questions, such as *what artifacts should be combined (across platforms, servers, etc.) to achieve a certain goal* and *how do we find artifacts that are semantically similar or connected*. In this vision paper, we are closing this gap by proposing a federated data science platform, called KEK [1], which addresses these neglected questions to break down silos in data science (DS).

Achieving this vision begins with the need to find, combine, and reuse artifacts as they are currently locked away in silos. There is no well-defined way of sharing these artifacts enhanced with semantic descriptions or even general metadata, neither much within a given data science platform and definitely not across multiple platforms. Thus, data scientists cannot automatically find relevant datasets and build a new pipeline on top of related ones since there is no way to identify them. As a practical use case and example, let us consider the case of reproducing experimental results of published articles, and analyzing insights driven from datasets.

Example. The problem is illustrated in Figure 1 – Laboratory 1 has a pipeline in a Java-based machine learning library (MLLib) operating on Dataset 1 to produce insights after enriching Dataset 1 with a local dataset; while Laboratory 2 has a pipeline in a Python machine learning library (Sklearn) that operates on Dataset 2 to produce insights described in a recent paper. At a *semantic* level, Dataset 2 could be joined with Datasets 1 and 3. Similarly, the pipelines are *semantically* equivalent; albeit in different programming languages and libraries. Yet, neither laboratory has any way to understand exactly what has been accomplished in the scientific community with respect to the datasets available at a specific data portal, e.g., Data Portal 1.

Existing data science platforms, such as MLFlow [49] and AutoML [12], tend to expand silos by locking-in pipelines and driven insights with limited or no collaboration support to force scientists to use the same platform. While a number of data science portals already exist, such as OpenML [41] and Kaggle[23], they still expect each user to load all open datasets, pipelines, and insights into their specific platforms – even before users can collaborate. Access to this community effort should not be restricted to a limited set of APIs, as in Kaggle. A more flexible mechanism to allow sharing of **datasets**

[1] KEK is the initials of the authors' first name. Kek means "raiser up of the light" in ancient Egypt.

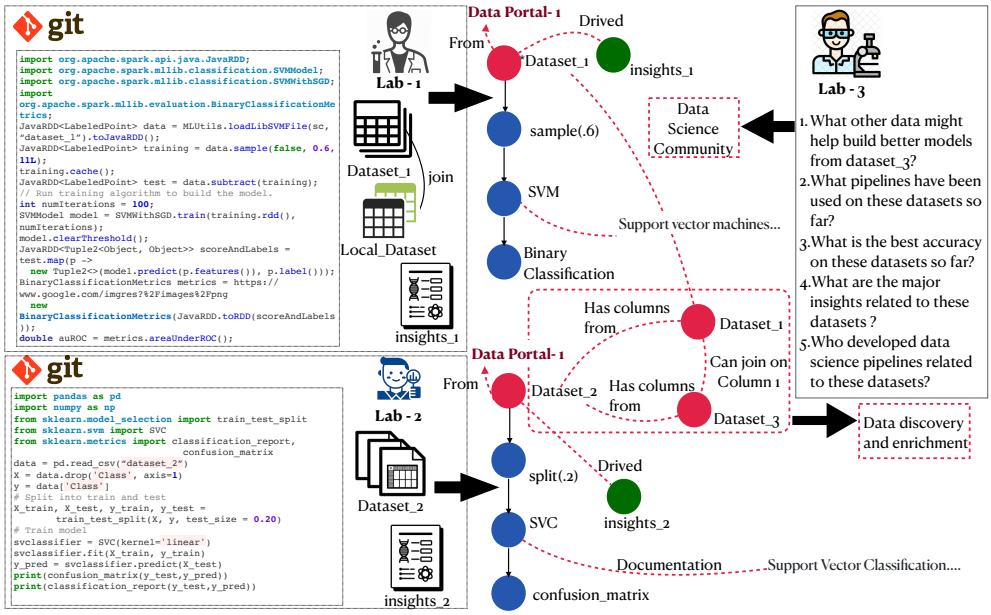


Figure 1: An overview of data science (DS) experimentations suffering from silos of data, pipelines, and insights. These silos prevent communication among the DS community and lead to consuming more time in data preparation, authoring pipelines, and finding insights related to datasets. The required automation to break down silos is denoted in red color.

and their associated **data science** artifacts is needed.

KEK therefore aims to provide a mechanism for the scientific community to discover and learn from each other's work automatically. In particular, KEK will help (i) discover and extract relevant data, (ii) enable scientists to collaborate more effectively regardless of the DS platforms they use, (iii) support efficient discovery of the most recent insights related to a dataset, (iv) enable scientists to reuse and combine (parts of) existing DS pipelines in novel ways, (v) enable reproducibility of experimental results with ease, and (vi) encourage innovative applications to automate several aspects of DS based on the most recent DS experimentation.

One of the key concepts to enable this vision and overcome silos is to abstract from syntactical differences of existing platforms and instead focus on the semantics of datasets, artifacts, and pipelines. Once we understand the semantics, we can more easily identify similar or matching artifacts and combine them in a federated manner. Instead of creating yet another silo by limiting KEK to a non-flexible standard, another key consideration is to retain a maximal degree of flexibility by capturing metadata and semantics in a flexible graph format. In our example from Figure 1, for instance, each laboratory's artifacts (stored in databases, file systems, or from a GitHub repository) are represented and indexed by an abstract graph representation that can be shared with other laboratories as illustrated in Figure 2.

We present an architectural overview of KEK in Section 2. Section 3 discusses how KEK could be used in practice. We discuss the research gaps for reaching our vision in Section 4, and related work in Section 5. Section 6 concludes the paper.

2. THE KEK PLATFORM

KEK aims to break up data silos by extracting and representing semantic information about data and artifacts in a flexible graph structure. The nature of extraction in KEK therefore results in a set of labeled graphs that together form decentralized data science knowledge graphs (**DSKG**s). KEK manages DSKGs using RDF-based knowledge graph technology because (a) it already includes the formalization of rules and metadata using a controlled vocabulary for the labels in the graphs ensuring interoperability, (b) it has built-in notions of modularity in the form of named graphs, so for instance, each laboratory's specific project could get its own named graph, (c) it is schema-agnostic, allowing the platform to support reasoning and semantic manipulation, e.g., adding new labelled edges between equivalent artifacts, as the platform evolves, and (d) it has a powerful query language with federated support (SPARQL) [3].

The KEK platform consists of four main sub-systems, as illustrated in Figure 3, and provides support for federated data science: (i) extracting semantic information from data items (datasets, pipelines, insights, artifacts, etc.), (ii) discovering links and similarities among data items at different granularities, such as datasets, tables, and pipelines, (iii) decoupling the semantics of experimentation on data items (pipelines and insights) from the used data science platform, (iv) interlinking these semantics with the relevant datasets, (v) processing complex queries efficiently in geo-distributed settings, (vi) synchronize the local DSKG with local datasets and scripts of pipelines at scale.

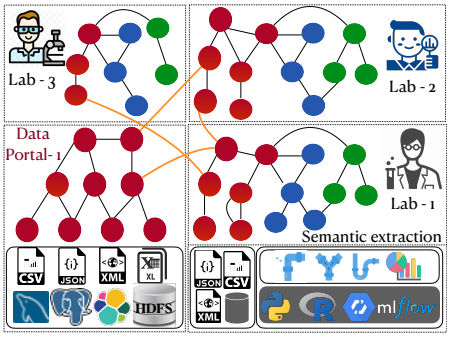


Figure 2: In KEK, decentralized data science knowledge graphs interconnect datasets to relevant pipelines and insights.

DSKG Management. In KEK, the DSKG construction sub-system profiles local datasets to construct a knowledge graph interconnecting data items, e.g., datasets, tables, and columns, accessed locally. The sub-system also maintains DSKG with the semantics captured and extracted from scripts of pipelines and insights. The data owner uses KEK to publish the graph to be accessible via the Web. In KEK, the DSKG services index local datasets and pipelines and maintain up-to-date local graphs capturing the extracted semantics.

KEK Federated Services. KEK provides federated services over geo-distributed DSKGs to allow automatic discovery and learning from data science projects across multiple data science users and heterogeneous data sources. A key feature of these services is to create and maintain links between decentralized DSKGs via, for example, link prediction. Another feature is a query processor that performs federated queries over the local knowledge graph and multiple other KEK portals to help scientists find and join datasets, pipelines, etc.

KEK Interface Services. KEK is designed to support interoperability with existing data science platforms and enable effective communication with data scientists. Thus, KEK provides API libraries to enable different data science platforms to communicate with KEK portals. In addition to structured queries over DSKGs, KEK supports natural language questions that help users easily find answers to their questions and extract the required information directly. A KEK portal is a RESTful server that accepts HTTPS calls.

KEK Foundations. To enable automatic learning from DSKGs, KEK harnesses a broad range of ML approaches including Graph Neural Networks (GNNs) [47] to support different functionalities, such as semantic data enrichment and pipeline automation. Our vision of KEK leverages parallelization and computation sharing to efficiently enable analytical workloads.

3. KEK IN USE

To avoid the dependency to a central instance or authority, KEK is envisioned as a federated platform of independent KEK portals, as shown in Figure 2. Organizations, such as enterprises, countries, or research labs, can then deploy their own instances of a KEK portal on top of their data lake. KEK offers a unique way for organizations to maximize data science potentials by capturing and learning from the usage and interdependencies of their data science artifacts including datasets, pipelines, and derived insights. Researchers, data scientists, and ML engineers, can deploy a KEK portal to capture the semantics of their pipelines and insights and use the KEK functionality to access artifacts shared by remote KEK portals. Hence, the KEK functionality could be implemented by different systems to run on private or public servers. Moreover, cloud providers can provide KEK portals as a service with varying degrees of reliability, performance, and security.

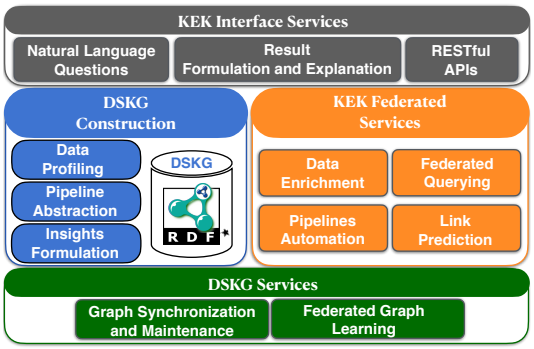


Figure 3: The KEK platform architecture.

Bootstrapping. When a new KEK portal wants to join, the first step is to use the *DSKG Construction* component (Section 4.1) to analyze the locally available data items, capture provenance, etc. and build a local DSKG covering datasets, processes, pipelines, and insights. The next step is to use the KEK *Federated Services* (Section 4.2) to "connect" the local DSKG to the ones from other KEK portals as illustrated in Figure 2.

Maintenance. As data scientists work on their projects and ideas, new datasets, pipelines, insights, etc., are continuously created. Hence, KEK portals need to regularly update their DSKG using the Construction components (Section 4.1) as well as *DSKG Services* (Section 4.3). Since this naturally affects the relationship to data items at other KEK portals, the information about the updates are shared, and the DSKG updated using the KEK Federated Services (Section 4.2).

Users of the KEK Platform. Different types of users interact with the system in different ways using the KEK Interface Services (Section 4.4). An administrator, for instance, might need a slightly different interface than a regular user who might prefer to use a natural language interface. Executing a user request in general can then easily entail using all other KEK components illustrated in Figure 3. As a concrete example, a researcher might want to work with a new dataset. Using the KEK infrastructure, it will be possible to find similar or joinable datasets as well as conclusions derived from similar datasets along with the pipelines that were used in

the process. Hence, given a specific task, users can use KEK to explore and propose potential analyses that have been used in similar cases. For data-driven journalism, given some desired insight, the KEK infrastructure can help find supporting datasets and pipelines.

4. RESEARCH CHALLENGES

This section highlights the open research challenges and opportunities of KEK's components.

4.1 DSKG Construction

In KEK, there is a need for novel methods to capture the semantics of a data science pipeline and its driven insights while interlinking the captured semantics with relevant datasets. As in other efforts in the search domain (e.g., schema.org) to specify a common vocabulary, one could leverage vocabularies to conceptualize relationships. Our DSKG includes nodes of different types, such as table, column, function, method, insight, and pattern. Some examples for edge types are: I) semantic similarity and inclusion dependency to interlink different data nodes, II) flows and reads to interlink code nodes together or to the used data nodes, and III) measure or aggregate to interlink insights related data nodes. We support automated or semi-automated maintenance of vocabularies to retain maximum flexibility.

Data Profiling: KEK data profiling aims at breaking down available artifacts into data items (columns, tables, datasets, pipelines, insights, etc.) to identify similarities and relationships. To achieve this goal, we will use the latest state of the art in data profiling and machine learning. KEK, for instance, requires the identification of hierarchies and statistics between data items such that this information can be used to construct a highly interconnected graph representation, in which vertices represent data items while edges represent relationships between them, such as "similarity". This graph is further annotated with provenance/metadata information and semantics to arbitrary domains of interest.

There is significant work in mapping columns and tables to concepts in knowledge graphs; but much of the work is primarily based on columns with string datatypes. More recent work has targeted numerical columns (e.g., [26]) but work of this nature is still at a fledgling stage. Our DSKGs are deductive graphs that utilize machine learning as well as inference rules to incrementally introduce and enhance the relationships among the different nodes in the graph. Therefore, the local DSKG will eventually be highly interconnected. This helps our profiling and construction process to scale to vast datasets.

Pipelines Abstraction: Similar programs are written with different APIs and languages. Initial efforts have been made to abstract the semantics of programs using static and dynamic program analysis techniques to extract language-independent representations of data science pipelines [2, 5]. Similar efforts capture the provenance of workflows, such as noWorkflow [33]. The example graph in Figure 1 (generated using [2]) illustrates how data flows through specific API pipeline calls, such as *SVM* or *SVC*. A key challenge that remains however is how one might recognize similar pipelines across frameworks or languages. There are many aligned benchmarks, such as CodeNet [34], that can be used by statistical models, such as Transcoder [37], to understand similarity across programs. One could leverage the associated natural language descriptions for APIs (e.g., documentation, forum posts) to generalize across multiple languages and frameworks. In Figure 1, for instance, the similarity of *SVC* and *SVM* could be derived from text, although this is still clearly an open challenge. Another challenge is to build multi-language independent abstractions for languages, that go beyond abstracting syntax trees. Systems, such as PROGRAML [7], derive abstract program graphs from neural models. These systems show initial promise for the development of language independent abstractions.

Insights Formulation: Data scientists use sophisticated libraries, such as R, Python, or Gnuplot, and tools, such as Tableau, Infogram, or Google Charts, for creating scripts capturing deeper insights from the data. While there are systems that have been proposed for extracting insights from an analysis of the data [9], they do not actually mine existing scripts targeting exploratory data analysis (EDA). Scripts targeting EDA are not easy to search; neither is it straightforward to enable automatic learning on them. There is a need for innovative approaches to capture the semantics of insights from the scripts, combined with comments in the scripts and connect them to their output including insights, observations, etc. Once this is accomplished, derived insights become searchable and processable at scale.

4.2 KEK Federated Services

The DSKG Construction analyzes the locally available datasets and scripts to build a local DSKG. The next step is to use the Federated Services to "connect" the local DSKG to the ones from other KEK portals via link prediction, as illustrated in Figure 2. We support federated querying, data enrichment, and pipeline automation on top of the decentralized DSKGs.

Link Prediction on DSKGs: In DSKGs, vertices represent data nodes, such as a node of type dataset, table, or column, or programming nodes, such as classes, functions, or methods, while edges represent relationships between these nodes, such as content similarity or function usage, respectively. We detect links between data items, such as tables or columns, using different methods, such as measuring content similarity. However, there are still other types of nodes or subgraphs, e.g., a pipeline or insights, where we need to predict links among them. We solve this problem as

a link prediction problem for knowledge graph completion using GNN-based models [50, 22]. KEK portals work transparently to interconnect different DSKGs and annotate DSKGs with provenance/metadata information. In KEK, learning the embeddings automatically is even more challenging due to the annotations in DSKG, i.e., hyper-relational facts [17], and the federated setup, which requires developing effective representation learning for datasets and data science artifacts in a geo-distributed environment.

Federated Querying and Exploration: Building upon knowledge graphs and existing standards, a variety of graph databases, commercial and research prototypes, is already available with basic support of federated querying. The challenge does not only lie within optimizing query execution across several KEK portals but also to keep each single one of them responsive despite potentially high query loads. Furthermore, KEK will support fine-grained and non-blocking query execution to produce results progressively. Thus, our federated execution model efficiently enables knowledge graph exploration and supports graph analytics queries generated by components, such as the semantic data enrichment and pipeline automation.

Semantic Data Enrichment: In the data preparation stage, data scientists tend to generate, in many cases, structured data, e.g., Dataframes, even from data sources of unstructured or semi-structured datasets, such as data logs or JSON documents. Usually, modeling results show data scientists that there is a need to add supplementary information to enrich the prepared dataset, as these dataframes may cover a limited number of cases. KEK assists users to easily extract relevant data, as discussed in Section 3.4. Moreover, KEK supports semantic data enrichment to find unionable, joinable, combinable data items, discover shortest paths, and schema integration. Users will be able to review discovered data before making the final decision on how to combine and further refine them. KEK further introduces functionalities to learn from the structure of DSKGs and make automatic recommendations for data enrichment based on semantic and syntactic matching.

Pipeline Automation Across Platforms: KEK's DSKG is able to capture API calls within a program, annotated with function calls and links to the used datasets. For pipelines, KEK does not join, i.e., combine two pipelines together. Instead, KEK interlinks similar pipelines to enable automatic graph learning for problems, such as pipeline automation as discussed in [20]. A DSKG takes the form of a knowledge graph and can be used in combination with deep graph generation networks [29] to model and generate pipelines for unseen datasets based on different representation learning techniques [47]. Then, we use state-of-the-art hyperparameter optimization systems, such as FLAML [43] or Auto-SKLearn [16], to recommend multiple optimized pipelines, see [20] for more details. Our model could be used by different ML platforms via KEK APIs to identify similar datasets to the unseen ones to generate new pipelines. Hence, KEK will provide a breakthrough for pipeline automation across platforms, i.e., by relying on the DSKGs, to help data scientists build data science pipelines quickly. There is a research opportunity to utilize the relevant datasets and previous analytical tasks to filter and classify generated pipelines.

4.3 DSKG Services

Graph Synchronization: KEK is not a static platform. As data scientists work on their projects and ideas, new datasets, pipelines, insights, etc., are continuously created. KEK platforms need to provide support to synchronize the local DSKG with local datasets and scripts of pipelines. This needs to incrementally maintain the DSKG and support pipelines generated by different platforms. This poses a research opportunity to develop a mechanism that efficiently updates the extracted semantics across scripts generated by different platforms.

Federated Graph Learning: KEK aims at developing a federated graph learning mechanism to learn graph representations (embeddings) across multiple DSKGs. KEK tasks, such as pipeline automation and semantic enrichment, benefit from this mechanism. We compute local and global features that generate embeddings based on the local and global DSKGs structure and topology. The graph features can be computed via analytical graph queries. Our federated graph learning is a promising technique to learn directly from the graph structure via sharing nodes' embedding with other remote connected nodes. This represents an open challenge for a scale message-passing framework in federated settings, and poses a research opportunity to develop an engine supporting variant embedding techniques for semantic queries [1]. This engine has to optimize the semantic query execution pipeline, automatically opt for the near-optimal embedding techniques, and estimate the cost of using this specific technique.

4.4 KEK Interface Services

For non-technical users, KEK provides question answering over DSKGs, automatically decide a data model for formalizing the results, and generate explanations.

Natural Language Questions: It is essential to reduce the technical effort required to explore and extract data/code from multiple KEK portals. Mapping a natural language question (NLQ) to a formal query language is challenging due to the ambiguity and multiple interpretations w.r.t. vertices related to data items, pipelines, and insights. Existing systems need thousands of annotated questions, such as NSQA [25], or require excessive preprocessing, such as such as gAnswer [21]. The preprocessing complexity is proportional to the KG size.

DSKGs are massive decentralized graphs that are fre-

quently updated. Thus, existing systems are impractical as the model should be re-trained from scratch for each update. There is a need for a model incrementally updated or trained independently of the graph. Thus, there is a need to develop a question answering system trained independently of the DSKG, as demonstrated by KGQAn [31]. The KGQAn system transforms a question into semantically equivalent SPARQL queries via a three-phase strategy based on natural language models trained generally for understanding and leveraging short English text. This poses a research opportunity to query multiple geo-distributed DSKGs and support natural language code and pipeline search [13].

Results Formulation and Explanation: Our methodology will develop different methods to estimate the query results' accuracy and index the NLQ segments and their relevant nodes and edges. The index will enhance the semantic understanding and linking of new NLQs based on the seen queries. The models will help in ranking query results. KEK's interface services should support data extraction in different formats based on the context of a given task and the NLQ semantic. For example, a data scientist may look for "Metro stations in Montreal," "Politicians born in New York City," or "Pipelines predicting car accidents in Aalborg". The result is not restricted to only one data model, e.g., a table format in the SQL language.

The result of these questions could be formalized as a map, table, or control flow graph, respectively. This represents an open challenge for adaptive models to predict the optimal formulation of results, e.g., as a table, graph, or map. Moreover, we need to annotate the results of NLQ with an explanation. Our methodology will adjust the query result's data model based on the NLQ semantics and its relevant data elements. This data model will include data explanations to help a data scientist understand the results in the context of a given task.

5. RELATED WORK

KEK is an end-to-end platform that enables the data science community to automatically discover, explore, and learn from existing data science artifacts and related datasets. The vision behind KEK is independent from or complementary to systems, such as Agora [39] or Cerebro [27], which focus on more technical aspects of executing data science pipelines across platforms, such as better utilization and unification of multiple computing resources or managing data as assets for trading, such as DMMS [15]. KEK, in contrast, is operating on a higher level of abstraction and could be built on top of the technical solutions provided by these systems.

In KEK, scripts of pipelines, and insights are managed by platforms of the user's choice. KEK captures the semantics of these scripts. Different tools, such as Vizier [4] and Ursprung [38], support the reproducibility of ML pipelines. The users can utilize these tools to manage their scripts without affecting KEK. LabBook [24] uses crowd sourcing to create a centralized knowledge graph to manage metadata about people, scripts and datasets, but KEK automatically extracts connections, in a highly distributed setting. AutoSuggest [48] is a tool helping in auto-completing a data-preparation pipeline. KEK focuses on modeling the detected insights and interlinking them with relevant datasets and pipelines. This will help automate several aspects of data science pipelines. Thus, these tools could benefit from KEK's knowledge graphs.

Systems, such as Google's Dataset Search Engine [30] and Helix [11], enable search over metadata of available datasets. Data discovery systems construct navigational data structures in the form of a linkage graph, such Aurum [14], an RDF knowledge graph, such as KGLac [19], or a hierarchical structure, such as RONIN [32]. Data sketches [28] can identify identical datasets used in different environments but cannot identify semantically similar data items or abstract a pipeline. Unlike these systems, KEK captures and extracts semantics of datasets, pipelines, and insights to construct a knowledge graph for data science enabling better collaboration in the community.

Multiple data versioning tools aim to track changes in the data used in ML models to enable reproducibility. Some tools were designed as S3 or Git extensions, such as Quilt [36], DVC [10], QRI [35], DataLad[8], and Git-LFS [18], to handle large data files. These tools do not handle schema changes, which may lead to breaking the execution of data science pipelines. Model management systems, such as ModelDB [42] and MLFlow [49], focus on reproducibility and tracing the modeling of experiments by capturing performance metrics, such as hyper-parameter and other values used in training. These data/model versioning tools do not capture the semantic abstraction of datasets and data science pipelines as proposed by KEK to enable advanced discovery and automatic learning.

6. CONCLUSION

KEK is a paradigm shift for open data science which brings together various communities, encourages more data scientists to share their work, and in doing so breaks down silos. In KEK, we utilize knowledge graph technologies to decouple the semantics of data science artifacts, e.g., pipelines and insights, from the data science platforms used to create and execute them. In doing so, KEK helps finding semantically similar artifacts and also finding out which artifacts should be combined to achieve a certain goal. The development of KEK poses numerous open research challenges that require innovative methodologies such as learning from decentralized knowledge graphs managed by geo-distributed KEK portals. In addition, new benchmarks are needed to mimic different workloads in federated data science.

7. REFERENCES

[1] H. Abdallah, D. Nguyen, K. Nguyen, and E. Mansour. Demonstration of KGNet: a cognitive knowledge graph platform. In *ISWC*, 2021.

[2] I. Abdelaziz, J. Dolby, J. P. McCusker, and K. Srinivas. A Toolkit for generating code knowledge graphs. *CoRR*, https://arxiv.org/abs/2002.09440, 2020.

[3] I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga, and P. Kalnis. Lusail: A system for querying linked data at scale. *PVLDB*, 11(4), 2017.

[4] M. Brachmann, W. Spoth, O. Kennedy, B. Glavic, H. Mueller, S. Castelo, C. Bautista, and J. Freire. Your notebook is not crumby enough, replace it. In *CIDR*, 2020.

[5] J. P. Cambronero and M. C. Rinard. AL: autogenerating supervised learning programs. *Proc. ACM Program. Lang.*, 3(OOPSLA):175:1–175:28, 2019.

[6] Canada Data Portal. https://open.canada.ca/.

[7] C. Cummins, Z. V. Fisches, T. Ben-Nun, T. Hoefler, and H. Leather. ProGraML: Graph-based deep learning for program optimization and analysis. *CoRR*, https://arxiv.org/abs/2003.10536, 2020.

[8] DataLad. http://www.datalad.org.

[9] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *SIGMOD*, 2019.

[10] DVC. https://dvc.org.

[11] J. B. Ellis, A. Fokoue, O. Hassanzadeh, A. Kementsietsidis, K. Srinivas, and M. J. Ward. Exploring big data with Helix: Finding needles in a big haystack. *SIGMOD Rec.*, 43(4), 2014.

[12] Fei-Fei Li and Jia Li. Cloud AutoML: Making AI accessible to every business. *GOOGLE CLOUD*, 2018.

[13] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou. CodeBERT: A pre-trained model for programming and natural languages. In *EMNLP*, 2020.

[14] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker. Aurum: A data discovery system. In *ICDE*, 2018.

[15] R. C. Fernandez, P. Subramaniam, and M. J. Franklin. Data market platforms: Trading data assets to solve data problems. *PVLDB*, 13(11):1933–1947, 2020.

[16] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter. Auto-Sklearn 2.0: Hands-free AutoML via meta-learning. *CoRR*, https://arxiv.org/abs/2007.04074, 2020.

[17] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann. Message Passing for Hyper-Relational Knowledge Graphs. In *EMNLP*, 2020.

[18] Git-lfs. https://git-lfs.github.com.

[19] A. Helal, M. Helali, K. Ammar, and E. Mansour. A demonstration of KGLac: A data discovery and enrichment platform for data science. volume 14, 2021.

[20] M. Helali, E. Mansour, I. Abdelaziz, J. Dolby, and K. Srinivas. A scalable AutoML approach based on graph neural networks. *CoRR*, https://arxiv.org/abs/2111.00083, 2021.

[21] S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *TKDE*, 30(5):824–837, 2018.

[22] M. K. Islam, S. Aridhi, and M. Smail-Tabbone. A comparative study of similarity-based and GNN-based link prediction approaches. *CoRR*, https://arxiv.org/abs/2008.08879, 2020.

[23] Kaggle Portal. https://www.kaggle.com/.

[24] E. Kandogan, M. Roth, P. M. Schwarz, J. Hui, I. G. Terrizzano, C. Christodoulakis, and R. J. Miller. Labbook: Metadata-driven social collaborative data analysis. In *BigData*. IEEE, 2015.

[25] P. Kapanipathi, I. Abdelaziz, S. Ravishankar, and et al. Leveraging abstract meaning representation for knowledge base question answering. *CoRR*, https://arxiv.org/abs/2012.01707, 2020.

[26] U. Khurana and S. Galhotra. Semantic annotation for tabular data. *CoRR*, https://arxiv.org/abs/2012.08594, 2020.

[27] A. Kumar, S. Nakandala, Y. Zhang, S. Li, A. Gemawat, and KabirNagrecha. Cerebro: A layered data platform for scalable deep learning. *CIDR*, 2021.

[28] J. Lemiesz. On the algebra of data sketches. *PVLDB*, 14(9), 2021.

[29] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. W. Battaglia. Learning deep generative models of graphs. *CoRR*, http://arxiv.org/abs/1803.03324, 2018.

[30] N. Noy, M. Burgess, and D. Brickley. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *WebConf*, 2019.

[31] R. Omar, I. Dhall, N. Sheikh, and E. Mansour. A Knowledge Graph Question-Answering Platform Trained Independently of the Graph. In *ISWC*, 2021.

[32] P. Ouellette, A. Sciortino, F. Nargesian, B. G. Bashardoost, E. Zhu, K. Pu, and R. J. Miller. RONIN: data lake exploration. *PVLDB*, 14(12), 2021.

[33] J. a. F. Pimentel, L. Murta, V. Braganholo, and J. Freire. NoWorkflow: A tool for collecting, analyzing, and managing provenance from python scripts. *PVLDB*, 10(12), 2017.

[34] R. Puri, D. S. Kung, G. Janssen, W. Zhang, G. Domeniconi, V. Zolotov, J. Dolby, J. Chen, M. R. Choudhury, L. Decker, V. Thost, L. Buratti, S. Pujar, and U. Finkler. Project CodeNet: A large-scale AI for code dataset for learning a diversity of coding tasks. *CoRR*, https://arxiv.org/abs/2105.12655, 2021.

[35] QRI. https://qri.io.

[36] Quilt. https://github.com/quiltdata/quilt.

[37] B. Rozière, M. Lachaux, L. Chanussot, and G. Lample. Unsupervised translation of programming languages. In *NeurIPS*, 2020.

[38] L. Rupprecht, J. C. Davis, C. Arnold, Y. Gur, and D. Bhagwat. Improving reproducibility of data science pipelines through transparent provenance capture. *PVLDB*, 13(12), 2020.

[39] J. Traub, Z. Kaoudi, J. Quiané-Ruiz, and V. Markl. Agora: Bringing together datasets, algorithms, models and more in a unified ecosystem [vision]. *SIGMOD Rec.*, 49(4), 2020.

[40] USA Data Portal. https://www.data.gov/.

[41] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, pages 49–60, 2014.

[42] M. Vartak and S. Madden. MODELDB: opportunities and challenges in managing machine learning models. *IEEE Data Eng. Bull.*, 41(4):16–25, 2018.

[43] C. Wang, Q. Wu, M. Weimer, and E. Zhu. FLAML: A fast and lightweight automl library. In *MLSys*, 2020.

[44] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[45] World Health Organization data portal. https://www.who.int/data/gho.

[46] World Trade Organization data portal. https://data.wto.org/.

[47] Z. Wu, S. Pan, F. Chen, and et al. A comprehensive survey on graph neural networks. *The IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020.

[48] C. Yan and Y. He. Auto-Suggest: Learning-to-recommend data preparation steps using data science notebooks. In *SIGMOD*, 2020.

[49] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, F. Xie, and C. Zumar. Accelerating the machine learning lifecycle with mlflow. *The IEEE Data Engineering Bulletin*, 41(4):39–45, 2018.

[50] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *CoRR*, https://arxiv.org/abs/2010.16103, 2020.

INODE: Building an End-to-End Data Exploration System in Practice

Sihem Amer-Yahia
CNRS, University Grenoble Alpes, France

Georgia Koutrika
Athena Research Center, Greece

Martin Braschler
Zurich University of Applied Sciences, Switzerland

Diego Calvanese
Free University of Bozen-Bolzano, Italy

Davide Lanti
Free University of Bozen-Bolzano, Italy

Hendrik Lücke-Tieke
Fraunhofer IGD, Germany

Alessandro Mosca
Free University of Bozen-Bolzano, Italy

Tarcisio Mendes de Farias
Swiss Institute of Bioinformatics, Switzerland

Dimitris Papadopoulos
Infili, Greece

Yogendra Patil
CNRS, University Grenoble Alpes, France

Guillem Rull
SIRIS Academic, Spain

Ellery Smith
Zurich University of Applied Sciences, Switzerland

Dimitrios Skoutas
Athena Research Center, Greece

Srividya Subramanian
Max Planck Institute for Extraterrestrial Physics, Germany

Kurt Stockinger
Zurich University of Applied Sciences, Switzerland

ABSTRACT

A full-fledged data exploration system must combine different access modalities with a powerful concept of *guiding* the user in the exploration process, by being *reactive* and *anticipative* both for data discovery and for data linking. Such systems are a real opportunity for our community to cater to users with different domain and data science expertise.

We introduce INODE - an end-to-end data exploration system - that leverages, on the one hand, Machine Learning and, on the other hand, semantics for the purpose of Data Management (DM). Our vision is to develop a classic unified, comprehensive platform that provides extensive access to open datasets, and we demonstrate it in three significant use cases in the fields of Cancer Biomarker Research, Research and Innovation Policy Making, and Astrophysics. INODE offers sustainable services in (a) data modeling and linking, (b) integrated query processing using natural language, (c) guidance, and (d) data exploration through visualization, thus facilitating the user in discovering new insights. We demonstrate that our system is uniquely accessible to a wide range of users from larger scientific communities to the public. Finally, we briefly illustrate how this work paves the way for new research opportunities in DM.

1. INTRODUCTION

The Data Management (DM) community has been actively catering to Machine Learning (ML) research by developing systems and algorithms that enable data preparation and flexible model learning. This has resulted in several major contributions in developing ML pipelines, and formalizing algebras and languages to facilitate and debug model learning, as well as designing and implementing algorithms and systems to speed up ML routines [23]. Conversely, existing work that leverages ML for DM [25] is nascent and covers the use of ML for query optimization [14] or for database indexing [13]. This paper makes the case for democratizing *Intelligent Data Exploration* by leveraging ML for DM.

Traditionally, database systems assume the user has a specific query in mind, and can express it in the language the system understands (e.g., SQL). However, today, users with different technical backgrounds, roles, and tasks are accessing and leveraging voluminous and complex data sources. In many scenarios, they are only partially familiar with the data and its structure, and their user information needs are not well-formed. In such settings, *expanding traditional query answering to data exploration*

is a natural consequence and requirement and with it comes the need to redesign systems accordingly. This need translates to several challenges at different levels.

(*Interaction*). Regarding interaction with the system, the biggest challenge is to enable the user to express her needs through a variety of **access modalities**, ranging from SQL and SPARQL to natural language (NL) and visual query interfaces, that can be used and intermingled depending on the user needs and expertise as well as the data exploration scenario. The second challenge is that of **user guidance**, i.e., users should be allowed to provide *feedback* to the system, and the system should leverage that feedback to improve subsequent exploration steps.

(*Linking*). Once a user need has been formulated and sent to the system, a search is executed over a (fixed) data set. Users may be aware which additional data sets could be of interest. However, they do not always know how to correctly link, integrate, and query more than one data source to generate rich information. This introduces the challenges of **data linking**, so that new data sources can be added to the system, as well as **knowledge generation**, so that queries over unstructured data can be supported. Both of these aim at enabling the continuous expansion of the "pool" of available data sources, thus making more data available to users.

(Guidance). Traditionally, the system will return to the user a set of tuples that concludes the search. There is a lot of work on how to improve performance for query workloads (predict future queries, build indices adaptively, etc.), but still the system has a rather passive role: anticipating or at best trying to predict the next query and then optimize its performance accordingly. Hence, the challenge of **system proactiveness** arises. The output is not only the set of results but also recommendations for subsequent queries or exploration choices. In our vision, the system guides the user to find interesting, relevant or unexpected data and actively participates in shaping the query workload.

In a nutshell, a full-fledged data exploration system must *combine different access modalities with a powerful concept of guiding the user* in the exploration process. It must *be reactive and anticipative*; co-shaping with the user the data exploration process. Finally, while data integration has been around for a while, *the ability to tie together data discovery and linking is a central question in an intelligent data exploration system.*

(Evaluation). An essential part of our proposal is the development of an evaluation framework to enable the end-to-end assessment of an intelligent data exploration system. This requires to formalize system metrics and human metrics that are necessary for data linking and integration, multi-modal data access, guidance, and visualization.

Related Work. Several systems address components of our vision. A number of them address interaction by enabling NL-to-SQL [3], SQL-to-NL [12] or both [11] (see a summary in [1]). Recommendation strategies can be leveraged to guide users [17]. Work on interactive data exploration aims at helping the user discover interesting data patterns based on an integration of classification algorithms and data management optimization techniques [6]. Each of the above-mentioned systems tackles specific data management challenges as so-called *insular solutions*. However, these insular solutions have not been integrated to tackle the end-to-end aspect of intelligent data exploration targeted at a wide range of different end users.

Combining all the challenges above requires an elaborated system whose multi-aspect behavior and functionality is the result of a synergy between disjoint technologies, and integrates them into a new ensemble. This gives rise to multiple approaches that vary in computational complexity, and raises new challenges that can benefit from recent advances in ML.

In summary, this paper makes the following contributions: One-size-does-not-fit-all when building a full-fledged data exploration system. For instance, the exploration operators are not all the same across different domains since exploring health data requires different semantics than exploring galaxies. Our aim is to encapsulate that semantics in higher level constructs, e.g. exploration by example, by natural language and by recommendation. Similarly, our aim is to build the components necessary for a full fledged system. We illustrate the need for intelligent data exploration with relevant use cases (Section 2). We describe INODE[1], a system that we are currently building as part of a project funded by the European commission (Section 3). To fully complete our vision, we provide open research challenges to be addressed at the intersection of DM and ML (Section 4).

2. USE CASES

In this section, we describe two of our three use cases - cancer research and astrophysics - and show how INODE can tackle them. The system is targeted for domain scientists as well as the general

[1]http://www.inode-project.eu/

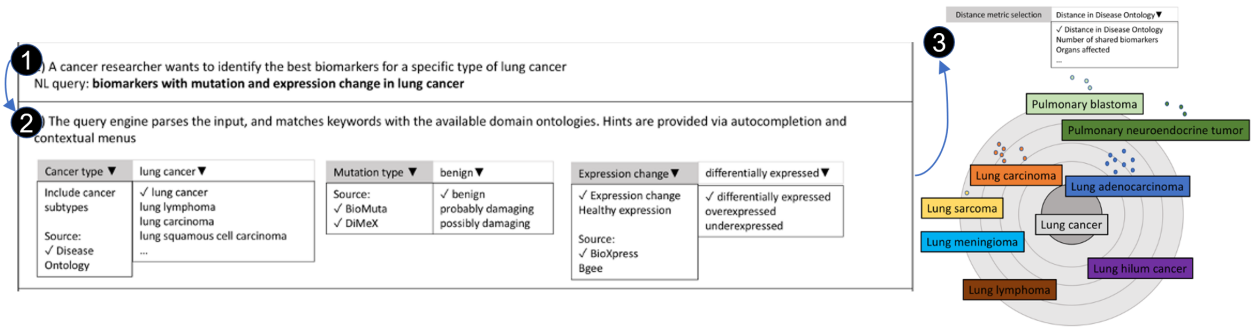


Figure 1: Natural language query interface with user assistance. Step 1: User enters a natural language query. Step 2: System parses query and matches keywords against the available ontology to enable term disambiguation; the user iterates the process. Step 3: System visualizes various cancer types that are similar to lung cancer. The distance metric between the diseases can be chosen by the user, e.g. by semantic distance.

public.

Use Case 1: Cancer Research (Natural Language and Visual Data Exploration). Fred is a biologist who studies cancer. His goal is to find which specific biomarkers are indicators for a certain type of lung cancer. He needs natural language exploration.

INODE offers support for NL queries, query recommendations, and interactive visualizations triggered by NL queries (see Figure 1). For instance, Fred starts with **a request in NL** for the topics related to lung cancer but is not sure how to continue after inspecting the results. INODE steps up and **recommends different options**: to expand the search using experimental drugs for treating lung cancer, or to focus on a subset of lung cancer types associated with a certain gene expression. Fred chooses to **expand his search** to one of the recommended topics, and receives a new list of lung cancers, drugs and genes. Additionally, INODE **explains in NL** how results are related. That helps him in selecting experimental drugs for certain gene expressions. After a few such queries, the system **visually analyzes the results** for Fred to study. Fred learns about the similarity between different types of cancer based on distance metrics that he can choose. In order to enable such data exploration, several different databases need to be integrated and potentially be correlated with findings from research papers.

Use Case 2: Astrophysics (Exploration with SQL-Pipelines). In the era of big data, astronomers need to analyze dozens of databases at a time. With the ever increasing number of publicly available astronomical databases from various astronomical surveys across the globe, it is becoming increasingly challenging for scientists to penetrate deep into the data structure and their metadata in order to generate new scientific knowledge. Sri, an astrophysicist, explores astronomical objects in SDSS, a large sky survey database [2]. Sri would like to examine Green Pea galaxies, first discovered in a citizen science project called 'Galaxy zoo', that recently gained attention in astronomy as one of the potential sources that drove cosmic reionization.

Figure 2 shows a sequence of three consecutive processes of analyzing astrophysics data. Sri relies on selected examples at each step and requests to see comparable ones. In the first query, she **asks to find galaxies with similar colors** as Green Pea galaxies. She then **requests objects with similar spectral properties, like emission line measurements, star formation rates etc.,** as those returned in the first step. The last query finds **similar galaxies in terms of their relative ratios and strength of emission lines**. As a result, Sri discovers that green pea emission line ratios are similar to high redshift galaxies.

INODE guides any user in making such new discoveries in an intuitive simpler way, without having to write complicated SQL queries or perform manual analysis of thousands of galaxies. For instance, INODE helps a user **choose among similarity dimensions** rather than rely on her ability to provide them. Additionally, INODE shows to the user **alternative queries** to pay attention to, thus increasing the chances of making new discoveries.

Crucially, INODE can be extended with additional resources which requires close interaction with domain scientists. Detailed user guides are in preparation.

[2]https://www.sdss.org/

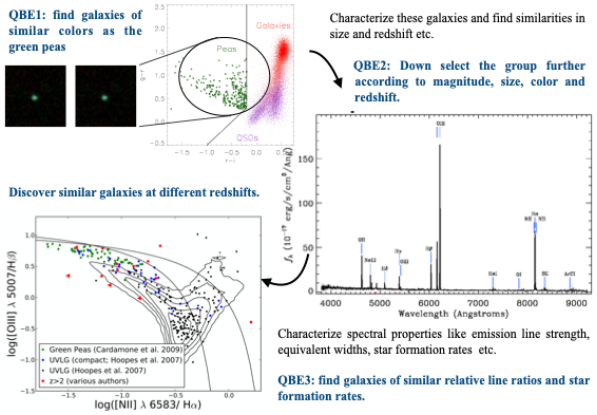


Figure 2: Exploring astrophysics data.

3. CURRENT INODE ARCHITECTURE

The main novelty of INODE is bringing together different data management solutions to enable *intelligent data exploration* (see Figure 3). Although some of these solutions and research challenges have been tackled previously, they have not been combined into such an end-to-end intelligent data exploration system, which in turn opens up new research challenges.

INODE's major components are as follows: (1) *Data Modeling and Linking* enables integration of both structured and unstructured data. (2) *Integrated Query Processing* enables efficient query processing across federated databases leveraging ontologies. (3) *Data Access and Exploration* enables guided data exploration in various modalities such as by natural language, by recommendation, by example or visually. We refer to these as *operators*.[3] Even though INODE is an integrated system, each of the components can also be used independently. The system is targeted for domain scientists as well as the general public.

3.1 Data Modeling and Linking

This component links loosely coupled collections of data sources such as relational databases, graph databases or text documents based on the well-established *ontology-based data access* (OBDA) paradigm [26]. OBDA uses a global ontology (knowledge graph) to *model* the domain of interest and provides a conceptual representation of the information in the data sources. The sources are linked to elements in the global ontology through declarative *GAV mappings* [15]. It is well-known that designing OBDA mappings manually is a time-consuming and error-prone task. The Data Modeling and Linking component of INODE aims at automatizing this task by providing two mechanisms: *data-driven* and *task-driven mapping* generation.

Data-driven Mapping Generation. This mechanism deals with linking novel data sources to the system. The idea is to rely on *mapping patterns* that describe well-assessed and sound schema-transformation rules usually applied in the design process of relational databases. By analyzing (driven by the patterns) the data sources, it is possible to automatically derive a so-called *putative ontology* [20] describing both the explicit entities and relationships constituting the schema and the implicit ones inferrable from the data. From the mapping patterns, one can also automatically derive mappings that link the data sources to the putative ontology.

Task-driven Mapping Generation. This mechanism is applied whenever a task or a query is formulated that uses specific *target* ontology elements that are not yet aligned with the putative ontology. In such scenario, the semantics of the query are used to automatically generate mappings to align the target ontology with the putative ontology.

Knowledge Graph (KG) Generation. This service transforms unstructured information hidden in large quantities of text (e.g. repositories of scientific papers) to an exploitable structured representation through an NLP pipeline. INODE follows an Open Information Extraction (OIE) approach to convert each sentence of the corpus into a set of relational triples, where each triple consists of a subject, an object, and a predicate (relationship) linking them. We leverage a number of preprocessing techniques, including co-reference resolution and extractive summarization to improve the quality of the extracted relational triples. We combine different OIE methods (rule-based, analytics-based and learning-based) to achieve both high precision and high recall [19, 22]. The relational triples are further linked with domain-specific ontology concepts before being integrated into the knowledge graph.

Note that all tasks of the *Data Modelling and Linking* component are executed offline and hence do not require interactivity.

3.2 Integrated Query Processing

This component is responsible for the execution of queries using Ontop [28], a the state-of-the-art OBDA system. Ontop allows the users to formulate queries in terms of *concepts* and *properties* of

[3] A prototype implementation of the major system components can be found at: `http://www.inode-project.eu/opendatadialog/`

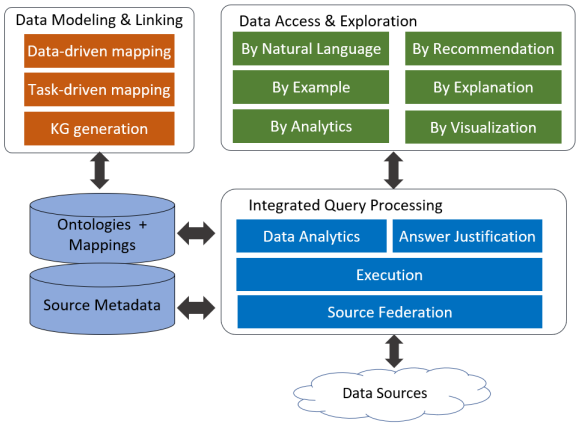


Figure 3: Major components of the INODE architecture.

their domain of expertise (represented in *knowledge graphs*), rather than in terms of table and attribute names used in the actual data sources. Hence, users do not have to be aware of the specific storage details of the underlying data sources in order to satisfy their information needs.

Query Execution. This service provides on-the-fly *reformulation* of SPARQL queries over the domain ontology to SQL queries over the data sources. An approach based on reformulation has the advantage that the data available in the data sources does not need to be duplicated in the query processing system, but can be kept in the data sources as-is. This means that the Query Execution service is guaranteed even in the common scenario where the user does not own the data nor does have the right to copy them. To produce reformulations that can efficiently be executed over the data, in INODE we use optimization techniques such as *self-join elimination for denormalized data* [28] and *optimizations of left-joins* [27].

Source Federation. The Source Federation service deals with distributing the processing of queries over the available data sources. INODE provides *seamless federation* over the SQL data sources.

In seamless federation, users send queries against a unified view of the remote endpoints without the need to be aware of the actual vocabularies used in the federated endpoints. The challenge is to automatically detect to which sources which components of the query need to be dispatched, to collect the retrieved results, and to combine them into a coherent answer. We address this challenge by relying on the knowledge about the sources encoded in the OBDA mappings. Note that in a seamless setting, the end-user interacts with the endpoint as usual, and remains unaware of whether the system will perform a federated query to retrieve the answers. Given that efficiency is a crucial requirement, in this, mostly interactive setting, our approach requires a dedicated cost-model able to minimize the number of distributed joins over the federation layer, in order to favor more efficient joins at the level of the sources.

Data Analytics. The data analytics service exploits novel and efficient query reformulation and optimization techniques [28] to compute complex analytical functions. Such techniques are based on algebraic transformations of the SPARQL algebra tree, rather than on Datalog transformations as traditionally done in the OBDA literature. This shift of paradigm allows for an efficient implementation of analytical functions such as SPARQL aggregates. It is worth noting that INODE, through Ontop, provides the first open-source reformulation-based system able to support SPARQL aggregates.

3.3 Data Access and Exploration Operators

We describe the set of operators currently available individually within INODE.

Exploration by Natural Language. For translating a natural language question into SQL or SPARQL, INODE uses *pattern-based*, *graph-based* and *neural network-based approaches*. For translating from NL to SQL, INODE extends the pattern-based system SODA [3] with NLP techniques such as lemmatization, stemming and POS tagging to allow both key word search queries as well as full natural language questions. In addition, we use Bio-SODA [21], a graph-based system to enable NL questions over RDF graph databases.

Finally, INODE integrates the *neural network-based approach* ValueNet, which leverages transformer architectures to translate NL to SQL [4]. The ultimate goal of INODE is to combine all these techniques into an *intelligent hybrid approach* that improves on the errors of each of the individual systems.

Exploration by Explanation. One of the biggest hurdles in today's exploration systems is that the system provides no explanations of the results or system choices. Nor does the system trigger input from the user, for example, by asking the user to provide more information. In INODE, we enable a conversational setting, where the system can (a) ask for clarifications and (b) explain results in natural language. This interaction assumes that the system is capable of analyzing and understanding user requests and generating its answers or questions in

natural language.

One approach used in INODE builds on *Template-based Synthesis* [12]. This approach considers the database schema as a graph and a query as a subgraph. We use templates that tell us how to compose sentences as we traverse the graph and we use different traversal strategies that generate query descriptions as phrases in natural language. Furthermore, to generate NL descriptions that use the vocabulary of a particular database, INODE enriches its vocabulary by *leveraging ontologies* built by the Data Modeling and Linking components. To further improve INODE's explanation capabilities, we are working on an approach to automatically learn templates, which is especially critical for databases with no descriptive metadata, such as SDSS. Essentially, we are using *neural-based methods* to translate from SQL or SPARQL to natural language.

Exploration by Example and by Analytics. By example is a powerful operator that encapsulates multiple semantics. It takes a set of examples, such as galaxies or patients, and explores its different facets, filters them, finds similar/dissimilar sets, finds overlapping sets, joins them with other sets, finds a superset, etc. Additionally, by-example operators can be combined with by-analytics to find sets that are similar/dissimilar wrt some value distributions.

By-example and by-analytics operators can be represented in the *Region Connection Calculus* (RCC) [16] and are, in their general form, computationally challenging. For instance, by-subset is akin to solving a set cover problem, which has been extensively studied [5]. Similarly, by-join requires to have appropriate indices. In INODE, we adopt two approaches. One is based on a *r*elational backend in which individual operators are translated into SQL. The other one is an *i*n-memory Python implementation that relies on pre-computing and indexing sets.

Exploration by Recommendation. In a mixed-initiative setting, the system actively guides the user in what possible actions to perform or data to look at next. In INODE, we are interested in recommendations in both *cold-start* (where the user has not given any input) and *warm-start* settings (where the user has asked one or more queries but may not know what to do next). In the former case, the goal is to show a set of example or starter queries that the users could use to get some initial answers from the dataset (e.g. [9]). In the latter case, the system can leverage the user's interactions (queries) to show possible next queries (e.g., [8]).

A big differentiator is the availability of query logs. In case no query logs are available, the system should still provide recommendations. In INODE we are addressing the recommendation problem from different angles, i.e., generating recommendations: (a) based on *data analysis* [7] (b) by *NL-based processing and query augmentation techniques* leveraging knowledge bases (c) by *user log analysis*.

Exploration by Visualization. In information retrieval, search queries result in a list of candidates ranked by their matching score [18]. This also holds true for INODE, as most exploration operators generate multiple potential answers. However, results are not individual items such as documents, but data sets (i.e. sets of items) and have to be communicated to the user differently to support their goals. Not only do users have to decide, which data set contains the answer they are looking for, but also to compare the results, to assess redundancies, discrepancies and other surprising or interesting differences in order to draw hints on how to continue the exploration. The goals of the *by-visualization* data access and exploration interface are two-fold: (1) Enable "explorers" to understand, compare and decide based on the provided results and (2) enable them to interact with the results by enabling indirect query manipulation, identifying and highlighting parts that are of interest for further analysis and guiding them towards interesting regions [24].

Our processes for user requirements elicitation confirms our goals stated above and is based on the *User Centered Design* standard [10]. In addition to that, users emphasized the importance to compare differences as well as similarities of queries and results. As a baseline, we enabled the visualization of multiple tables with direct manipulation capabilities and currently work on an *overview visualization* that spans the result data space.

4. CONCLUSIONS

A full-fledged data exploration system should *learn about data sources*, *learn about users and queries*, and leverage this knowledge to facilitate and *guide users*. All these challenges constitute new opportunities for ML research to contribute to DM which are elaborated in the extended version of this paper [2].

5. ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 863410.

REFERENCES

[1] K. Affolter, K. Stockinger, and A. Bernstein. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal*, 28(5):793–819, 2019.

[2] S. Amer-Yahia, G. Koutrika, F. Bastian, T. Belmpas, M. Braschler, U. Brunner, D. Calvanese, M. Fabricius, O. Gkini, C. Kosten, D. Lanti, A. Litke, H. Lücke-Tieke, F. A. Massucci, T. M. de Farias, A. Mosca, F. Multari, N. Papadakis, D. Papadopoulos, Y. Patil, A. Personnaz, G. Rull, A. C. Sima, E. Smith, D. Skoutas, S. Subramanian, G. Xiao, and K. Stockinger. INODE: building an end-to-end data exploration system in practice [extended vision]. *CoRR*, abs/2104.04194, 2021.

[3] L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger. Soda: Generating sql for business users. *PVLDB*, 2012.

[4] U. Brunner and K. Stockinger. Valuenet: A natural language-to-sql system that learns from database information. *ICDE*, 2021.

[5] G. Cormode, H. J. Karloff, and A. Wirth. Set cover algorithms for very large datasets. In *CIKM*, 2010.

[6] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. AIDE: an active learning-based approach for interactive data exploration. *IEEE Trans. Knowl. Data Eng.*, 28(11):2842–2856, 2016.

[7] A. Glenis, Y. Stavrakas, and G. Koutrika. Pyexplore: Clustering-based sql query recommendations. In *under submission*, 2020.

[8] M. L. Guilly, J. Petit, and V. Scuturici. SQL query completion for data exploration. *CoRR*, abs/1802.02872, 2018.

[9] B. Howe, G. Cole, N. Khoussainova, and L. Battle. Automatic example queries for ad hoc databases. In *SIGMOD*, 2011.

[10] International Organization for Standardization. ISO 9241-210:2019 - Ergonomics of Human-System Interaction — Part 210: Human-Centred Design for Interactive Systems, 2019.

[11] R. J. L. John, N. Potti, and J. M. Patel. Ava: From data to insights through conversations. In *CIDR*, 2017.

[12] A. Kokkalis, P. Vagenas, A. Zervakis, A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Logos: a system for translating queries into narratives. In *SIGMOD*, 2012.

[13] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In *SIGMOD*, 2018.

[14] A. Kristo, K. Vaidya, U. Çetintemel, S. Misra, and T. Kraska. The case for a learned sorting algorithm. In *SIGMOD*, 2020.

[15] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, 2002.

[16] S. Li and M. Ying. Region connection calculus: Its models and composition table. *Artificial Intelligence*, 145(1):121 – 146, 2003.

[17] J. Liu, Z. Zolaktaf, R. Pottinger, and M. Milani. Improvement of SQL recommendation on scientific database. In *SSDBM*, 2019.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[19] D. Papadopoulos, N. Papadakis, and A. Litke. A methodology for open information extraction and representation from large scientific corpora: The cord-19 data exploration use case. *Applied Sciences*, 10(16), 2020.

[20] J. F. Sequeda and D. P. Miranker. Ultrawrap Mapper: A semi-automatic relational database to RDF (RDB2RDF) mapping tool. In *Proc. of the 14th Int. Semantic Web Conf., Posters & Demonstrations Track (ISWC)*, 2015.

[21] A. C. Sima, T. Mendes de Farias, E. Zbinden, M. Anisimova, M. Gil, H. Stockinger, K. Stockinger, M. Robinson-Rechavi, and C. Dessimoz. Enabling semantic queries across federated bioinformatics databases. *Database*, 2019, 2019.

[22] E. Smith, D. Papadopoulos, M. Braschler, and K. Stockinger. Lillie: Information extraction and database integration using linguistics and learning-based algorithms. *Information Systems*, 2021.

[23] E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht. Keystoneml: Optimizing pipelines for large-scale advanced analytics. In *ICDE*, 2017.

[24] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, volume 33, pages 401–410. Wiley Online Library, 2014.

[25] I. Stoica. Systems and ML: when the sum is greater than its parts. In *SIGMOD*, 2020.

[26] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev. Ontology-based data access: A survey. In *IJCAI*, 2018.

[27] G. Xiao, R. Kontchakov, B. Cogrel, D. Calvanese, and E. Botoeva. Efficient handling of SPARQL optional for OBDA. In *ISWC*, 2018.

[28] G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E. Güzel-Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese, and E. Botoeva. The virtual knowledge graph system ontop. In *ISWC*, 2020.

How Inclusive are We? An Analysis of Gender Diversity in Database Venues

Angela Bonifati
Lyon 1 University & Liris CNRS
France
angela.bonifati@univ-lyon1.fr

Michael J. Mior
Rochester Institute of Technology
Rochester, NY
mmior@cs.rit.edu

Felix Naumann
Hasso Plattner Institute
University of Potsdam, Germany
felix.naumann@hpi.de

Nele Sina Noack
Hasso Plattner Institute
University of Potsdam, Germany
nele.noack@student.hpi.de

ABSTRACT

ACM SIGMOD, VLDB and other database organizations have committed to fostering an inclusive and diverse community, as do many other scientific organizations. Recently, different measures have been taken to advance these goals, especially for underrepresented groups. One possible measure is double-blind reviewing, which aims to hide gender, ethnicity, and other properties of the authors.

We report the preliminary results of a gender diversity analysis of publications of the database community across several peer-reviewed venues, and also compare women's authorship percentages in both single-blind and double-blind venues along the years. We also obtained a cross comparison of the obtained results in data management with other relevant areas in Computer Science.

1. INTRODUCTION

Increasingly, the computer science and database community are recognizing the importance of actively increasing diversity, in particular gender diversity among researchers, or removing impediments to the advancement of underrepresented researchers in the field. For instance, ACM SIGMOD and VLDB together started an initiative "to create an inclusive and diverse database community with zero tolerance for abuse, discrimination, or harassment", and the D&I in DB initiative coordinates such efforts across the data management community[1].

One opportunity to increase diversity might be double-blind reviewing, hiding the authors names and thus effectively hiding their gender from the reviewers. While there might be further signals about the gender of the author(s), for instance in their writing style or the topic of the paper, author names are the most direct indicators of gender to reviewers and readers. Moreover, gender lookup using names has also been adopted in recent work on the authorship of Computer Science contrasted with other fields of study [10]. Only in an (albeit large) minority of cases the gender is not unambiguously revealed by the first name even if the reviewer does not personally know the author and their gender. Other methods, such as targeted surveys in our community or image processing on photos of personal homepages, could be used to address these ambiguous cases. These methods go beyond the scope of our work and are the subject of future investigation.

In this study, we analyze and compare the participation of women in papers at various top-level conference and journals. To this end, we make use of a commercial service to assign gender based on first names for many languages. While we realize that gender is not a binary concept distinguishing women and men, we do not have the means to identify any more fine-grained designations based on the given data, which matches that which reviewers and readers usually have at their disposition. Next, we have downloaded and prepared reference data from DBLP. With our dataset, we are able to compare the evolution of such diversity across the years and compare the diversity across venues, some of which perform double-blind reviewing. Our analysis considers only accepted papers; we do not report about the diversity of rejections due to lack of data.

Our preliminary findings show that there is an overall growth of the number of accepted papers authored by women in major database conferences, with some slight differences. We also examined how the data management field stands with respect to other fields such as HCI, AI, Algorithms, Network-

[1] https://dbdni.github.io/

ing, and Operating Systems. In this landscape, the differences might also be due to the gender-composition of the researchers in the respective fields. Finally, we could not observe a tangible difference between single-blind and double-blind reviewing for the data concerning the SIGMOD conference. The analysis of the submission data could be enlightening in that case.

The following Section 2 discusses related work. Then, Section 3 introduces both our approach to identify the gender of authors and the considered publication datasets. Section 4 is the core of this empirical paper, presenting our analytical findings. Finally, we conclude with an outlook on possible further analyses in Section 5.

2. RELATED WORK

Snodgrass provides an excellent survey of literature analyzing the effects single- vs. double-blind reviewing [6], which we do not repeat in our empirical work here. Many studies from different research fields do mention gender fairness as a goal of double-blind reviewing. However, the cited results are often inconclusive: some report a significant bias, others do not observe this. Snodgrass concludes [6]: "These studies show that revealing author identity, specifically the gender of the author, can sometimes have an effect on acceptance rates."

In the database research field, the SIGMOD conference is a particularly interesting venue to analyze: until the year 2000 it employed single-blind reviewing before switching to double-blind reviewing in 2001. Apart from gender bias, the original impetus for this change, and for double-blind reviewing in general, is to avoid any bias of reviewers to more favorably review and to more readily accept papers by well-known, prolific authors, and to thus let the content speak for itself. We are not the first to analyze the effects of this change of reviewing policy. Madden and DeWitt identified "prolific" authors and their success rate at SIGMOD and VLDB conferences from 1995 until 2005 [3]. They conclude that "double-blind reviewing has had essentially no impact on the publication rates of more senior researchers in the database field". Tung performed a similar study on the same data, concluding "that there are indications that double-blind reviewing does have an impact in terms of papers accepted for famous people in SIGMOD" [9]. However, neither of the two works addresses gender diversity.

Tomkins et al. also analyzed the impact of double-blind reviewing using data from a single computer science conference edition: WSDM'17. Here, some reviewers had access to author information while others did not [7]. In their study they also analyze the "Matilda effect", in which "publications from male authors are associated with greater scientific quality, in particular if the topic is male-typed" [2]. Tomkins et al. [7] found no statistically significant impact on bidding and reviewing both for papers with a woman as first author and for papers with a majority of women as authors. They do perform a meta-analysis across seven studies, which, put together, show a statistically significant negative bias for these papers.

Other analyses of bibliometric data from DBLP-DB have been carried out in the past, e.g., to study the collaboration network in our community [1]. That study shows that there is a power law on the frequency of publications and presents other statistics, such as the number of co-authors per scholar. They do not discuss the impact of gender in this kind of analysis.

3. PREPARING, SELECTING, AND ANALYZING DATA

In this section, we explain how we selected and preprocessed the data used in our analysis. We also discuss how we carried out our assessment. Our entire analysis is reproducible and the source code along with additional results are publicly available[2].

3.1 Defining paper gender

In this paper, we focus on gender analysis of bibliographic data in the data management field. While other analyses could be done by considering diversity of the writing style, paper topics, or other factors, we do not consider them here. We focus on authorship information for a paper and define three different categories of gender when associating it to a paper.

- A paper whose first author is a woman (FAW)
- A paper whose last author is a woman (LAW)
- A paper with any author being a woman (AAW)

Clearly, papers that fit the first two definitions also fit the last definition, but not vice versa. These definitions are sufficient to let us take an initial dip into the analysis and study the trends of woman authorship in our community. We distinguish the three aforementioned definitions in our analysis and show and cross-compare the corresponding results.

[2] https://github.com/HPI-Information-Systems/GenderAnalysis/

Alternative definitions are clearly possible to study the data under different perspectives and by considering other dimensions in addition to gender. For instance, one can think of analyzing bibliographic data by looking at other diversity criteria, which are equally important, such as race, ethnicity, country of origin, culture, affiliation, (academic) age, etc. Although these criteria are applicable to our corpus, we do not regard them here.

3.2 First-name analysis

Automatically deriving gender from first names is known to be a very difficult problem [5]. Some rules of thumb might apply. For instance, knowing the gender of first names in case of familiarity with the language of the country of origin of that author makes sense as an applicable rule. However, in some spoken languages, there might exist ambiguity in the gender of first names. For instance, Andrea is typically a woman-identified first name in Germany, whereas it is exclusively a men-identified first name in Italy. The same first name is sporadically used for men in Germany for people being Italian immigrants. In these cases, the country of origin of the authors could help us disambiguate the gender of the authors. While it would be possible to use country of origin in the DBLP data in order to help disambiguate the names, this affiliation country data is quite sparse ($< 30\%$) and we decided not to use it in this first analysis.

From the list of publications, we infer the authors' full names and split them into first, (middle), and last names. For obtaining the genders of the first names, we use Gender API[3], a commercial online platform to determine gender by first names. In the first step, we use the list of first names to look up the gender. If the first name is abbreviated, we look to the middle name(s). For a given first name, Gender API provides the predicted binary gender along with an estimated accuracy and the number of samples of that name held in their database. We use the predicted gender if the accuracy is higher than 50% percent. Otherwise, we label the first name concerned with 'neutral'. There are also some names for which Gender API does not provide any result. We label these names as well as fully abbreviated ones with 'unknown'.

To not under-represent either men or women, we consider the gender of all unknown and gender-neutral names to randomly be either man or woman, based on the overall gender distribution in the portion of the data where the predicted gender is more certain. We are aware that this binary assignment does not respect all genders and that the extending the observed women/men distribution to all other names might introduce some bias. Furthermore, the name someone is given at birth may not necessarily be one that matches their gender identity. However, as our goal is to assess potential bias among reviewers, we expect the gender commonly perceived to be associated with a particular author's name to be a sufficient starting point for this analysis. We also tried alternative distributions, e.g., unknown gender data considered all men or all women, and observed that the overall trends of accepted papers for women did not change and no further insights could be gleaned from the obtained results.

3.3 Venue selection

Our data is taken from the DBLP computer science bibliography[4]. We downloaded the entire proceedings data available in DBLP for a selection of popular database research and other CS venues and collected all authorship information. Our analysis includes ACM SIGMOD, VLDB, ICDE, EDBT and CIDR conferences. Notice that among these, only SIGMOD is double-blind, while the remaining ones are single-blind. For the data concerning VLDB, we combined the conference data (VLDB) with the data from Proceedings of VLDB (PVLDB), the latter being the replacing journal starting from 2008. We label the combination as VLDB.

For comparison, we planned to also include other top database journals, such as VLDB Journal and ACM Transactions on Database Systems (TODS). Due to the low absolute number of papers appearing in TODS, we decided to dismiss it in the presentation of the results. Finally, we include a lower-ranked conference (DASFAA) and a lower-ranked journal (DKE) to allow a comparison between higher and lower ranked venues. Table 1 lists for each venue the years for which we gathered data, and the overall number of papers for that duration.

Venue	Years	# pubs	# authors
CIDR	2003 – 2020	476	1,173
DASFAA	1989 – 2020	1,939	4,220
DKE	1985 – 2020	1,719	3,438
EDBT	1988 – 2020	1,552	3,307
ICDE	1984 – 2020	4,743	8,046
SIGMOD	1975 – 2020	4,065	6,959
(P)VLDB	1975 – 2020	5,198	8,621
VLDBJ	1992 – 2020	907	1,996

Table 1: Captured years and number of papers for each conference

[3] https://gender-api.com/

[4] https://dblp.org/

Furthermore, we cross-compare the data in our community with neighboring communities in computer science. For that purpose, we regarded CS-Rankings[5], considered the data of selected fields, and chose the corresponding conferences listed there, as reported in Table 2.

Field	Venues	avg. authors
AI	AAAI, IJCAI	3.10
Algorithms	FOCS, SODA, STOC	2.44
Databases	SIGMOD, VLDB, ICDE, PODS	3.44
HCI	CHI, UIST, UbiComp, Pervasive, IMWUT	3.93
Networking	SIGCOMM, NSDI	4.20
Operating Systems	OSDI, SOSP, EuroSys, FAST, USENIX ATC	4.34

Table 2: Venues listed for other fields

4. DIVERSITY RESULTS

In this section, we report the results of our analysis concerning (i) papers authored by women accepted in the data management community across the years and venues listed in Table 1, and (ii) trends of accepted papers in neighboring communities in computer science for the fields and conferences listed in Table 2. Across all figures, we report a 3-year moving average percentage of papers following in each category.

Figures 1, 2 and 3 show this average for the three categories of the first (FAW), last (LAW), and any author (AAW) having a woman-identified name across all years in which that venue published papers. By looking at the results, we can observe the following:

- CIDR shows the lowest diversity across all categories, but, being a single-track conference and being a biannual event until recently, the overall number of papers is lower compared to other venues leading to low significance of our analysis. Moreover, the conference was limiting the number of papers submitted by the same author (to 1 or 2 depending on the years) and focusing solely on systems, vision, and prototype papers.
- For SIGMOD we created two regression lines: one up to 2000 for its single-blind process, and one from 2001 onward to reflect its double-blind process. We did not observe a remarkable difference in the percentage of accepted papers by women after shifting to a double-blind review policy. However, we cannot draw a conclusion on this aspect, as this would require inspecting more data (including the submission data).
- By examining the FAW results, we can observe a higher percentage of papers accepted in DASFAA, which could suggest that women as first authors are more successful in this conference. However, this trend is less prominent in the LAW and AAW results for DASFAA. A similar trend can be observed for DKE with peaks in the period 1995-2000 for all three percentages.

As a disclaimer for the results reported above, we let the reader notice that the outcome of our analysis should be taken with some caution. Indeed, the presence of authors with unknown genders for which we did infer the gender and the fact that we collapsed the entire proceedings into one bulk piece of data (without distinction between long and short papers with different respective acceptance rates) might lead to some confounding factors. As such, our analysis is preliminary and can certainly be improved in future work.

Finally, Figure 4 shows the results for the percentage of papers authored by women (FAW) aggregated per CS field. We chose five additional research fields as reported in Table 2.

From these results, we can observe that the HCI field sees the highest percentages of papers by women across the years, whereas the Operating Systems field is lowest. We can also see that, at least recently, the database field is faring somewhat better than the remaining fields. Nevertheless, these results should be taken with a grain of salt since they also depend on the gender composition of the various fields. In particular, we point out that the information about gender composition of the different fields is missing at present, as also highlighted in recent work on the dynamics of gender bias [4]. Once this information will be available, it can help interpret better the above results.

5. CONCLUSION

In this paper, we have focused on the gender impact on authorship in the data management area. We started from the assumption that women are an underrepresented group in computing [10, 11]. This assumption has been confirmed by the results of our study.

Our analysis was of course only a preliminary step towards many and more detailed analyses. For

[5] http://csrankings.org/

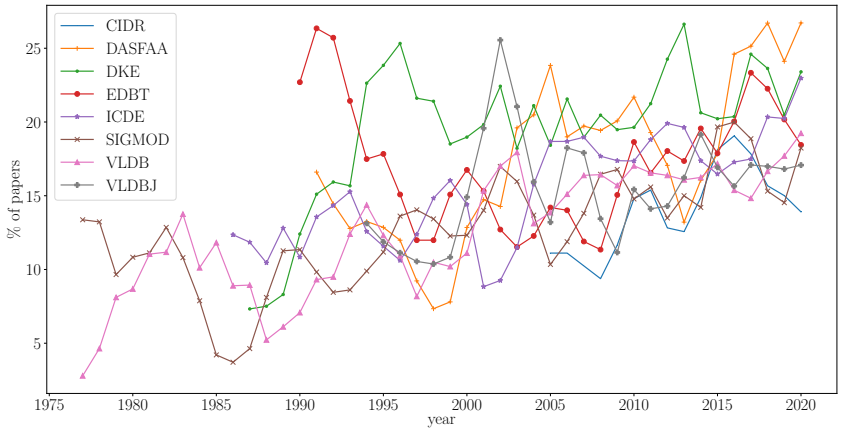


Figure 1: First author woman (FAW) percentages by year (3-year moving average)

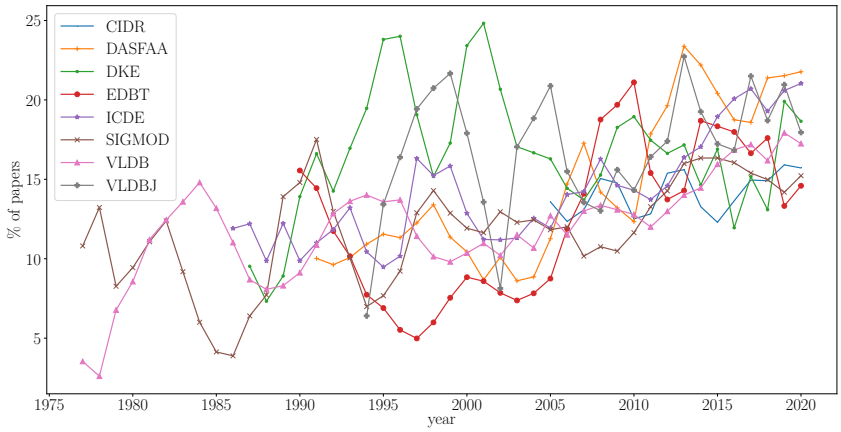


Figure 2: Last author woman (LAW) percentages by year (3-year moving average)

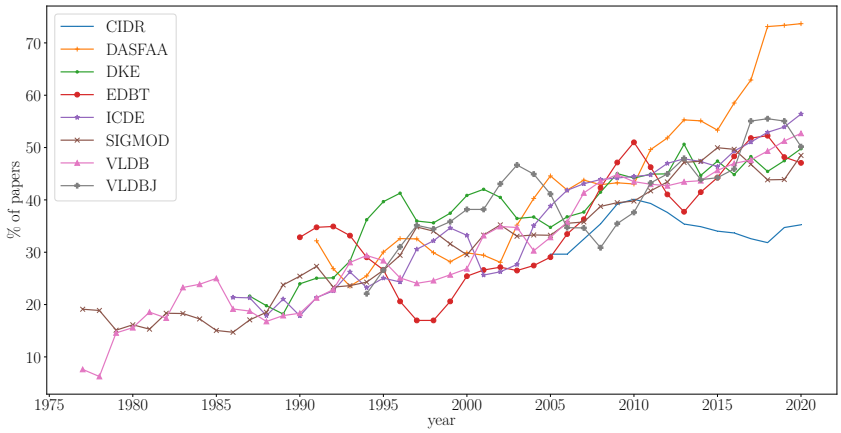


Figure 3: Any author woman (AAW) percentages by year (3-year moving average)

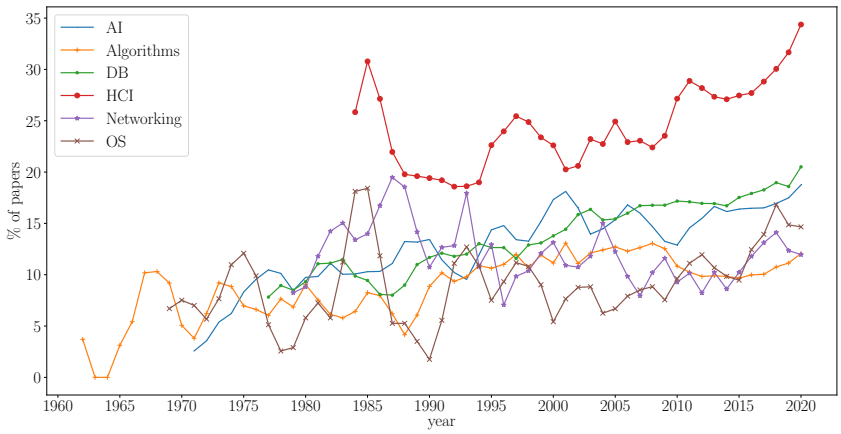


Figure 4: First author woman (FAW) percentages across fields in CS (3-year moving average)

instance, with affiliation data, the same statistics could be broken down by region, by country or by individual affiliation. Gender assessment using names can become more accurate by leveraging manual annotations and targeted surveys within our community or by image processing starting from website pictures, even if the latter has other limitations, such as solely considering gender as binary, the inherent noise of the available data, etc. Also, while the overall trends show an increase in diversity, it would be interesting to compare gender with the academic age to validate the hypothesis that this increase is mostly due to junior women entering the field.

An even more insightful analysis could be performed not only on accepted papers, as we do here, but including also data about submitted papers to the various venues. The latter would be more difficult, as it requires accessing sensitive data, such as the submission data and reviews for conferences and journals in our field. Moreover, this analysis would be applicable to one conference and one edition of the conference only, as it has been done for instance for the ICLR conference [8].

Acknowledgements

We would like to thank the colleagues who helped annotate first names: Mazhar Hameed, Hazar Harmouch, Lan Jiang, Ioannis Koumarelas, Nitisha Jain, Chao Zhang, Meihui Zhang, and Kyuseok Shim. We also thank Renée Miller for starting this discussion at VLDB 2020 and Arun Kumar for giving us inspiring and very helpful feedback.

6. REFERENCES

[1] Ergin Elmacioglu and Dongwon Lee. On six degrees of separation in DBLP-DB and more. *SIGMOD Record*, 34(2):33–40, 2005.

[2] Silvia Knobloch-Westerwick, Carroll J. Glynn, and Michael Huge. The Matilda effect in science communication: An experiment on gender bias in publication quality perceptions and collaboration interest. *Science Communication*, 35(5):603–625, 2013.

[3] Samuel Madden and David DeWitt. Impact of double-blind reviewing on SIGMOD publication rates. *SIGMOD Record*, 35(2):29–32, June 2006.

[4] Thomas J. Misa. Dynamics of gender bias in computing. *Communications of the ACM*, 64(6):76–83, 2021.

[5] Lucía Santamaría and Helena Mihaljević. Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science*, page 4:e156, 2018.

[6] Richard Snodgrass. Single- versus double-blind reviewing: An analysis of the literature. *SIGMOD Record*, 35(3):8–21, 2006.

[7] Andrew Tomkins, Min Zhang, and William D. Heavlin. Reviewer bias in single- versus double-blind peer review. *Proceedings of the National Academy of Sciences (PNAS)*, 114(48):12708–12713, 2017.

[8] David Tran, Alex Valtchanov, Keshav Ganapathy, Raymond Feng, Eric Slud, Micah Goldblum, and Tom Goldstein. An open review of OpenReview: A critical analysis of the machine learning conference review process. *CoRR*, abs/2010.05137, 2020.

[9] Anthony K. H. Tung. Impact of double blind reviewing on SIGMOD publication: A more detail analysis. *SIGMOD Record*, 35(3):6–7, September 2006.

[10] Lucy Lu Wang, Gabriel Stanovsky, Luca Weihs, and Oren Etzioni. Gender trends in computer science authorship. *Communications of the ACM*, 64(3):78–84, 2021.

[11] Telle Whitney and Valerie Taylor. Increasing women and underrepresented minorities in computing: The landscape and what you can do. *Computer*, 51(10):24–31, 2018.

We are starting a new series to provide advice to mid-career researchers. There are a number of programs that SIGMOD organizes for researchers at the beginning of their careers (PhD Symposium and the like) and senior people do not (or should not) need much help. There are considerable challenges for those who are about to transition from an early researcher to a more senior role. In academia, these are people who are about to get tenured that comes with starting to think of moving from shorter-term research objectives to longer-term ones. In industrial research, this corresponds to the transition from participating in projects to initiating and leading them. As a community we don't seem to talk about these challenges much. That is the gap this series attempts to fill. We will get the views of senior researchers from diverse backgrounds and diverse geographies. We will continue as long as we find original advice and the views are not repetitions.

M. Tamer Özsu
University of Waterloo

Congratulations! You Have Become a Senior Researcher. Now What?

Magdalena Balazinska
Paul G. Allen School of Computer Science & Engineering
University of Washington

It probably seems like yesterday that you were starting at your first post-PhD position, but with this latest promotion, whether it is tenure or promotion to a senior level at your company, you can no longer call yourself "junior". You are now stepping into the shoes of a senior researcher. Congratulations! This is a tremendous accomplishment, and you should celebrate. The road was long and often uphill. You finally made it.

Promotion to a senior role is a really great time to pause and reflect on where we are and where we would like to go next. We all know that seniority brings the opportunity to take even greater risks and have an even greater impact than the early years. That's in my opinion one of the greatest benefits of getting older (there has to be a positive side of getting wrinkles and having to attend more faculty meetings). It's important to take that opportunity and to continue working extra hard to do great things because, before long, the next promotion will loom on the horizon.

First and foremost, transitioning to a senior role is a great opportunity to grow as a *researcher*. To pause and ask: "What is the most important problem that I should be working on?" A senior researcher has already proven that they are very good at research. Now they get to define and assess excellence. They can define new research directions and even new research areas. They can explore unusual directions. In my case, as an assistant professor, I worried about making sure I had a steady stream of papers at top conferences (while also going through two pregnancies and raising little children), so I took the safe approach of building on existing open-source systems, such as Hadoop, for my projects. After promotion to associate professor, I embarked (with my colleagues) on an exciting project where we built our own big data system and cloud service, called Myria. I also started to explore unusual directions with my collaborators, such as how to price data or how to price cloud services, and more recently how to manage video data including 360-degree virtual reality videos. The ability to take greater risks let me take on more ambitious and more interesting projects than before and, in all cases, I was glad that I had chosen that path.

Seniority, however, opens much more than opportunities to grow as a researcher. It also opens more opportunities to expand the types of impacts one can have. Many senior researchers have start-ups based on their research. More junior researchers can also do that, but it's much easier as a senior researcher to get through the disruption to research caused by a new company. Some researchers decide to write a textbook, which requires great persistence and effort, but can have a major impact on how an entire subject area is taught. Other researchers, yet, apply for center grants or lead other large-scale initiatives or large-scale projects. In general, seniority implies the expectation of *leadership*

and much larger-scale impact. When one is recently promoted to a senior role, taking on tasks such as start-ups, books, or center leadership may still be a bit early. So this may not be something to do right away, but these options are things to consider and start thinking about. I took the approach of getting involved in and leading large-scale initiatives after tenure. As an associate professor, I was the Principal Investigator on an NSF IGERT (training) grant and led the development of a new program for data science education across the University of Washington (UW) through data science specializations, called "options", which are now offered in many units at the undergraduate, graduate, or both levels. This leadership work, together with a general deeper engagement in the UW data science institute, called eScience, put me in a good position to later, after my next promotion to full professor, become the director of eScience, and recently the director of the Paul G. Allen School of Computer Science & Engineering. Many paths for impact beyond research are available to senior researchers and it can be exciting, challenging, but also very rewarding to take on these different types of opportunities. As a recently promoted researcher, it's good to start thinking about such possibilities.

One aspect of seniority that is important but can easily be overlooked is the necessity to grow as a *mentor* for more junior researchers, especially those junior researchers who are underrepresented in computing. In our field these include women, people of color, people with disabilities, first generation college students, and others. A senior researcher must not think only about themselves and their own success. They cannot focus only on their direct reports or advisees. They must think about their broader team, their institution, their research area, even all of computer science as a field. We can all think back throughout our careers and remember great people who helped us along the way by providing advice, inviting us to give a talk, sending an opportunity our way, participating in a workshop we were organizing, etc. We can all remember being inspired by senior researchers describing entirely new research directions for the community. We can all remember being thankful to senior researchers for stepping up and arguing the importance of computer science as a field to higher level governments. As a senior researcher, it is now our turn to do the same. Everyone learns quickly to think and help their students and their immediate team. Senior researchers must do much more than that. While this may feel intimidating, it need not be. Helping others can be as simple as providing respectful advice to an assistant professor from another university who just gave a presentation, or asking a committee to pause when the shortlist for an important role or keynote talk only contains the small set of individuals who get invited to everything. It can take the form of helping to try new ideas for how we run our conferences, serving as PC chair, general chair, or other. It can mean participating in a national organization, in the US these would be organizations such as the National Science Foundation Computer and Information Science and Engineering advisory council, the Computing Community Consortium, the Computing Research Association, or other. There are many ways to lead a community once one recognizes that it is a senior researcher's responsibility and opportunity. When we get tenured and move into associate professor ranks, it's good to start thinking about these types of contributions, start preparing oneself to take them on, and slowly start to explore these types of leadership roles.

On a related thought, senior researchers also need to take on greater leadership roles and responsibilities *at their home institutions*. Senior researchers need to contribute to the vision, direction, and success of their organization. We work at fantastic companies and established universities. We benefited from their support and resources to establish ourselves. Now that we are senior researchers, we need to take our organizations to the next level and ensure their continued success. This can also take many forms. One can chair a major committee such as an admission or hiring committee. One can work on revamping some aspect of the organization whether related to education, research, or policies and procedures. One can start a major initiative that builds on the organization's strengths and enables some dramatic new fundraising. Whatever the approach, it is important to simply acknowledge and embrace the fact that our organizations are relying on us to significantly contribute as its senior members. As an associate professor, one approach to helping our universities in this way without becoming overwhelmed is to pick only one activity of this type and focus on it. Later, as a full professor, one can expand to leading multiple such activities.

While we embrace our senior role to take our career and impact to the next level, it's important to also remember *our community*: our partners, children, parents, friends, and neighbors are counting on us. We all know that life is short, and time goes by fast, but it can be surprising just how fast time goes. As an example, before the pandemic hit, I was planning to take my kids on a trip to Europe but didn't get around to it. They were too young, I thought. It's an expensive trip. Then the pandemic hit and now I'm realizing how few years I have left before my kids go off to college. At the same time, I look back fondly on all our ski outings, camping trips, violin recitals, soccer games, and other activities. I'm also proud of how well they are doing at school in spite of their learning differences and the school struggles that inevitably hit anyone who doesn't fit the mold. It can feel like a cliche, but a promotion to senior researcher is a really good time to pause and ask ourselves: "In addition to my exciting work, am I accumulating regrets in my personal life, or am I accumulating fond memories?" "Am I self-centered, or am I helping my family and community?" It's important to support our families and communities and do something meaningful outside of work.

Finally, while reaching a senior level can feel like one can sigh a sigh of relief, senior roles can also be very stressful. Between all the exciting projects, responsibilities, and challenges, we can get pulled in too many directions, and have to work non-stop. Everyone around us will say: "Remember to take care of yourself" but sometimes the question becomes "How can I do that with everything going on?" So let me leave you with three ideas that I learned much later than I wish I had. I hope you will find them helpful.

For stress, I learned from a colleague who is a professor of psychology that stress often comes from an imbalance between the demands that are put upon us and the resources we have to respond to those demands. For that reason, when stress becomes too high, it's good to share it with someone or look for extra resources in another way. For example, if there's a difficult situation at work and difficult decisions need to be made, can we find others to discuss the situation with and discuss the best response? If funding is challenging, writing grant proposals with others can be both more fun and less stressful. In case of a difficult situation at home, sharing the situation with our manager or department chair can help to identify options to perhaps reduce work responsibilities temporarily. When one is faced with too many community-serving or other tasks, perhaps that is a good time to delegate something to a more junior person who could benefit from the exposure and learn to take on more of these responsibilities; or perhaps it is a good time to recruit one or two fewer Ph.D. students. When stress arises, it's good to recognize the imbalance and ask ourselves: I have insufficient resources to meet the demands that are put upon me. How can I either reduce those demands or access additional resources?

For overall self-esteem, it's good to remember that everyone around us is an iceberg: We see the tip of the iceberg, which shows all the successes and awards, but we don't see the much larger bottom of the iceberg with all the challenges and struggles. For that reason, if you find yourself comparing your accomplishments to that of others, stop right there. The only valid comparison is yourself now to yourself last year and yourself in the future. We can all grow and do better. The question to ask ourselves is how do we want to grow in the next year? How do we want to do better?

And, finally, when something doesn't work out, when we make a mistake, when we fail in some way, it can be really helpful to say it out loud ("This really wasn't my finest moment"; "I really could have done a better job with X"; "I need to figure out how to do better with Y"), acknowledge that we need to let ourselves grieve over that failure, and then conclude by saying: "I didn't fail. I tried and found a way that doesn't work. Let me try differently next time and see if it works out better". Thomas Edison said just that: "I have not failed. I've just found 10,000 ways that won't work."

I hope you found some of the above helpful. And while it's good to listen to advice, after listening, one should always do what one thinks is right and not necessarily what the advice recommended. Enjoy your new senior researcher role. I hope it will enable you to do great things, both for you and those around you.

Accelerating Video Analytics

Joy Arulraj
Georgia Institute of Technology

MOTIVATION. The advent of inexpensive, high-quality cameras has led to a rapid increase in the volume of generated video data [19, 16]. It is now feasible to automatically analyze these video datasets at scale due to two developments over the last decade. First, researchers have designed complex, computationally-intensive deep learning (DL) models that capture the contents of a given set of video frames (*e.g.*, objects present in a particular frame [11]) [15]. Second, the computational capabilities of hardware accelerators for evaluating these DL models have increased over the last decade (*e.g.*, TPUs) [8]. We anticipate that automated analysis of videos will reduce the labor cost of analyzing video datasets in a wide range of important applications [14].

BACKGROUND. Motivated by these developments, researchers have recently proposed several novel video database management systems (VDBMSs) [2, 1, 9, 21, 4]. These systems accelerate declarative queries over videos using techniques like training a lightweight, specialized model to filter out irrelevant frames [12], or sampling a subset of important frames [10, 3]. The queries they support primarily focus on detecting objects of interest (*e.g.*, searching for frames containing atleast two cars in a surveillance video). To accelerate this query, the VDBMS may train a lightweight model to quickly filter out irrelevant frames that are unlikely to contain cars [12]. By reducing the number of invocations of the heavyweight oracle model (*i.e.*, the more accurate DL model specified by the user [12, 5]), the VDBMS speeds up the query with a tolerable drop in query accuracy.

CHALLENGES. State-of-the-art VDBMSs suffer from two limitations that constrain their utility and computational efficiency. First, these systems primarily focus on accelerating object detection queries over videos. So, they are not able to support queries associated with more complex vision tasks. For example, an important class of video analytics queries focuses on detecting and localizing *actions* – events spread across a sequence of frames (*e.g.*, "right-turn of a car") [17, 20, 6]. It is difficult to process such queries due to two reasons. First, current VDBMSs operate on individual frames (either using the lightweight filter or the heavyweight object detector). To detect an action, the VDBMS would need to identify features that span across multiple frames. Second, inference times of DL models tailored for action detection are higher than that of object detectors.

Another limitation is that it is computationally expensive for the VDBMS to train filters for each unique combination of: video content, oracle model, and predicate of interest. First, filters depend on video content (*e.g.*, day- vs night-time videos [18]). Second, the labels associated with the training frames are obtained using a specific model (*e.g.*, SSD [11]). Third, due to the limited capacity of filters, they are tailored for a specific predicate (*e.g.*, COUNT(CAR) > 2 [13]). These constraints increase the overall training cost associated with filters.

IDEAS. To tackle the first challenge, we will need to design novel algorithms for efficiently processing action queries. For instance, we could train a DL-based agent to quickly skim through video segments that are unlikely to contain the target action [7]. The agent would quickly generate proxy features of a given video segment and use them to choose the next video segment to process (*e.g.*, picking the resolution of the frames, the sampling frequency, *e.t.c.*) from a large space of possible segments. We anticipate that such task-specific optimizations will need to be developed for other vision tasks [4].

For the second challenge, it is important to develop unsupervised algorithms for sampling representative frames from a video. This will allow the VDBMS to answer ad-hoc queries using these representative frames instead of training a filter tailored for a specific predicate or oracle model. It is critical to obtain theoretical bounds on the likelihood of the representative frames satisfying the query accuracy constraint.

SUMMARY. An amalgamation of ideas in database systems, computer vision, and machine learning will help realize the vision of accelerating video analytics. We anticipate that VDBMSs will become more common in the future, and hence, the optimizations developed by the database community will become important.

REFERENCES

[1] BlazeIt. https://github.com/stanford-futuredata/blazeit.

[2] EVA. https://github.com/georgia-tech-db/Eva.

[3] J. Bang, P. Chunduri, and J. Arulraj. Eko: Adaptive sampling of compressed video data. *arXiv preprint arXiv:2104.01671*, 2021.

[4] F. Bastani, S. He, A. Balasingam, K. Gopalakrishnan, M. Alizadeh, H. Balakrishnan, M. Cafarella, T. Kraska, and S. Madden. Miris: Fast object track queries in video. In *SIGMOD*, pages 1907–1921, 2020.

[5] J. Cao, R. Hadidi, J. Arulraj, and H. Kim. Thia: Accelerating video analytics using early inference and fine-grained query planning. *arXiv preprint arXiv:2102.08481*, 2021.

[6] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139, 2018.

[7] P. Chunduri, J. Bang, Y. Lu, and J. Arulraj. Zeus: Efficiently localizing actions in videos using reinforcement learning. *arXiv preprint arXiv:2104.06142*, 2021.

[8] J. Dean, D. Patterson, and C. Young. A new golden age in computer architecture: Empowering the machine-learning revolution. *IEEE Micro*, 38(2):21–29, 2018.

[9] B. Haynes, M. Daum, A. Mazumdar, M. Balazinska, A. Cheung, and L. Ceze. Visualworlddb: A dbms for the visual world. In *CIDR*, 2020.

[10] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia. Noscope: Optimizing neural network queries over video at scale. *arXiv: Databases*, 2017.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[12] Y. Lu, A. Chowdhery, S. Kandula, and S. Chaudhuri. Accelerating machine learning inference with probabilistic predicates. In *SIGMOD*, pages 1493–1508, 2018.

[13] Y. Lu, A. Chowdhery, S. Kandula, and S. Chaudhuri. Accelerating machine learning inference with probabilistic predicates. In *SIGMOD*, pages 1493–1508, 2018.

[14] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, and A. Waldman-Brown. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[16] T. J. Sejnowski. *The deep learning revolution*. MIT press, 2018.

[17] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016.

[18] A. Suprem, J. Arulraj, C. Pu, and J. Ferreira. Odin: Automated drift detection and recovery in video analytics. *VLDB*, 13(11), 2020.

[19] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[20] H. Xia and Y. Zhan. A survey on temporal action localization. *IEEE Access*, 8:70477–70487, 2020.

[21] Y. Zhang and A. Kumar. Panorama: a data system for unbounded vocabulary querying over video. *VLDB*, 13(4):477–491, 2019.

Juliana Freire Speaks Out on Reproducibility and Hard Changes

Marianne Winslett and Vanessa Braganholo

Juliana Freire

https://vgc.engineering.nyu.edu/~juliana/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I am Marianne Winslett, and today I have here with me Juliana Freire, who is a professor at New York University. Juliana is an ACM Fellow, and she has a Google Faculty Research Award, an IBM Faculty Award, and an NSF Career Award. She is also the chair of SIGMOD, and her term of office ends in just a few days. Juliana's Ph.D. is from Stony Brook. So, Juliana, welcome!

Thank you so much, Marianne. Thank you so much for actually doing this great service to the SIGMOD community. I know for a fact that this series that you run is one of the most popular sections of SIGMOD RECORD. So, thank you so much for doing this.

It's a pleasure.

Your colleagues say that you have been quietly battling against outdated traditions in the database research community for a long time. What have been your biggest battles and biggest accomplishments there?

One thing that I've learned in all these years that I am in academia is that change is difficult, and change takes time. At SIGMOD, we wanted to make some big and some small changes. And some small changes actually turned out to be big. For example, to have a diversity of opinions, gender, geography, as well as cover more areas, we proposed changing the structure of the conference to have two co-chairs. We faced a lot of resistance. Now after two rounds of SIGMOD with two co-chairs, the feedback from the chairs, program committees, and authors has been overwhelmingly positive. So, I think that this is one example of a small thing that turned out not to be so small.

Another challenge has been to increase the adoption of reproducibility in our community. This is something that my colleague Dennis Shasha started in 2008. And we have been making baby steps since then. There is still a lot of work to be done on this front. But I guess we can talk more about this later. Yes?

Yes! Since Computer Science moves so fast, why do we need reproducibility?

It is precisely because Computer Science moves so fast that we need reproducibility. If I do some work and you want to build on my work, how are you going to do that if you cannot reuse and extend what I did? If you have to start from scratch, this is actually going to slow down Computer Science. Reproducibility is necessary, specifically to make it possible for science, and Computer Science to move forward.

Do you see circumstances where reproducibility might impact science in a negative way?

I can't think of how reproducibility can be bad for science. There are some barriers to reproducibility. For example, works and experiments that use private data or proprietary software can be difficult or impossible to reproduce. People also cite, for example, intellectual property as another barrier. More recently, there have been concerns about open science and reproducibility being misused by bad actors. So maybe that would be one potential negative side of reproducibility, but for society in general.

Should privacy, the right to be forgotten, factor into how reproducible science is maintained?

There can be privacy issues in the data that is used in a particular scientific result and that must be respected. But there are ways of mitigating this problem. There are people working on synthesizing datasets that have similar properties but that do not disclose personal information. If you are talking about privacy with respect to "Oh, I did my work, I published it at SIGMOD, I have my experiments, but I don't want anybody to see those experiments." then, I disagree because I think that science has to be open. If I have my results, in particular, if my research was funded by the federal government, it was paid by the taxpayers, I have no reason, no good excuse, not to actually make that available and open to everybody.

Another potential issue is that it's hard to keep a piece of software working in the long-term because the hardware underneath changes, the OS, and the libraries. Do we have a moral responsibility to keep our research artifacts working, and if so, for how long?

Yes, I think that this is an important topic of discussion, in particular because there are costs associated with this. Lots of people keep asking how much should we actually invest in keeping old work as opposed to funding new research that is going to lead to new results. I think that the new developments around virtual machines and all the infrastructure that we have right now with the cloud make it a lot easier to preserve these research artifacts – to increase their longevity and make them usable in the longer term. This is definitely easier now. We should not aim to have these artifacts living forever. But I think it's important to try and keep them, for as long as possible.

There are efforts that aim to preserve such artifacts. Software Heritage is an initiative, started by Roberto di Cosmo at INRIA, in France, that is collecting all pieces of software that have ever been produced in the world -- you can think of this as software archeology. The goal is to have them forever, whether they are going to be running forever, that's a different question.

If I am running in a modern environment and I want to build on top of something that is living in a virtual machine from the past, how do I do that?

Depending on what you want to do and what you need to do, it can be easy or hard. Nowadays, there are workflow systems that allow you to stitch together different virtual machines. So, if the work is self-contained and you just need to input something and get some output, that's trivial. If it requires modification to the code and integration with new libraries, then, it can be very difficult. But if you have the software, and ideally the source code, it may be possible to more easily adapt it than to build everything from scratch.

> ***[…] science has to be open. […] if my research was funded by the federal government, it was paid by the taxpayers, I have no reason, no good excuse, not to actually make that available and open to everybody.***

Provenance tracking, being able to tell what information a particular conclusion is based upon, is super important for scientists. Does it matter for other people?

Of course! Provenance and reproducibility are now applicable to everything. We are witnessing a data and computing revolution: everything that people do now in government, industry, and science is around data and computing. More and more, decisions are being made based on results and insights that are obtained from data and computations. Provenance is key, particularly if you are making important decisions that have serious consequences. You need to be able to know what you have done, and reason about what you have done to make sure that you can build trust in the results on which you base your decisions.

I think a great example of that would be that the CDC said the chance of catching the coronavirus outdoors was 10% or something like that. Journalists traced that fact back in the data and found out it was based on data from construction workers in Singapore. Being someone who's lived in Singapore for a long time, I can promise you they didn't understand what construction was about in Singapore. So, the conclusion they made was erroneous. But on the other hand, despite the fact that they could have traced it back, they probably would still have reached the same erroneous conclusion, wouldn't they?

I think that there is deeper issue here related to meta-studies**.** People collect data for different purposes and meta-studies attempt to combine them to synthesize new knowledge and draw their conclusions. The problem is that the context and assumptions that are made for each of the different projects and underlying data used in a meta-study can be different, and inconsistent. It's difficult to reconcile all of those, and I think that is what happened in the study you refer to. Because it's a construction site, but it was not necessarily enclosed. I think that was the issue, right?

I think now, the real issue which most people don't know is that the construction workers in Singapore live together in dormitories with like 12 people to a room in bunk beds. It is the closest packed environment that you can imagine. So, of course, the coronavirus is going to spread under those conditions. But they didn't think about that. They imagined that it was always caught at work.

But then, this is an instance where proper provenance was not actually captured. Because if we had correctly captured the contextual information where the data was actually gathered, you wouldn't have had that problem. But in practice, this is difficult to avoid. You cannot avoid all of these mistakes or oversights. This is why it is essential to have transparency and be able to trace back the steps. In this case, the journalists were able to go back and look at the data and figure the problem out. You need to capture as much provenance as you can to enable you and others to go back to assess and debug the results.

Your open-source workflow and provenance tracking system, VisTrails, was ahead of its time in many ways. What about its impact are you most proud of, and what lessons did you learn from that?

I think that VisTrails was my first project that had real practical impact. It ended up being widely used by many different people, different communities. Big projects adopted it. And there are lots of things that contributed to that. First, we had a great team working on the system. We had a group of Ph.D. students that were not only talented researchers, but that were also amazing hackers and very passionate about the project. VisTrails was written and rewritten about three or four times. And if you look at the code, it is professional. The system worked, and it worked well. An important lesson that I learned is that if you want to do something well, you

need to have the right team. And in this case, we were very lucky to have the dream team.

VisTrails is a good example of a multidisciplinary project. And for such projects to succeed, we also need to have the right collaborators. We were very fortunate to identify a number of people, including physicists, biologists, medical doctors, that worked closely with us and from whom we actually learned what the real problems were, what their real pains were. We designed a system to meet the scientists' needs. At the same time, because we were working so closely, not only did we solve their needs, but we also were able to get into a virtuous cycle: we solved the real problems that the scientists had, and at the same time, we found a number of interesting Computer Science problems. And this is how three different Ph.D. dissertations, and many papers, came out of the VisTrails system.

Another big challenge that we had was maintaining an open-source system at a university. Raising funds to support programmers (after the Ph.D. students were done) to actually keep the project going and supporting users is extremely challenging. We lack (both in funding agencies and at the universities) the proper infrastructure to keep research software engineers. This is a fight that I am still fighting within NYU. If we want to have successful Data Science, Computer Science applied to science projects, we need to have research engineers and proper career paths for them at the university – they are critical to the success of our research and need to be recognized as such.

At some point, you moved your focus from captured workflow to providing provenance support for Python scripts and Jupyter notebooks. Why is that?

This is another lesson that we learned from VisTrails. The project was very successful, but to use VisTrails and to reap up all the benefits that come from provenance that the system automatically collects, people have to adopt that system. There is not only a learning curve but also a ramp-up period in which you actually need to adapt your research environment and integrate it with VisTrails. For some people, that worked, but many people want to keep working with the tools that they are already familiar with. So, my vision was: "Can I get the same benefits of VisTrails, but within the environment of Jupyter, of Python, that tens of thousands of people actually use on a day-to-day basis?" Let's get reproducibility to the masses without having to put any burden on them.

That system, ReproZip, has shown itself to be really useful in creating reproducible artifacts. How did you come to develop ReproZip?

One of the key issues that we observed is that systems like VisTrails and other workflow systems, capture provenance for the steps that are followed in the workflows, for example, processing the data and building a machine learning model. If you have the specification, you can rerun those steps within the workflow system. The problem is that if I want to share them with you and you want to run those on your machine, you may not be able to because there are the dependencies, there are libraries, there are different Python versions, different scikit-learn[1] versions. ReproZip captures the provenance of your computational environment: everything that your experiment needs, files that it reads and writes, and libraries that it uses. It automatically creates a package that contains not only your computational steps but the whole environment required to run those steps. And once you have that, you can reproduce the experiment on a different machine or in different operating systems. ReproZip solves the dependency hell problem.

> ***[…] we need to have research engineers and proper career paths for them at the university – they are critical to the success of our research and need to be recognized as such.***

What if the artifact depends on old versions? Can you reproduce that?

Oh yes! ReproZip works as follows: when you run your experiment, it watches at the operating system's level everything that is touched and invoked by the experiment. If the experiment uses a specific Python library, ReproZip will identify the library. And when you create the package, ReproZip copies that library, the old library, into the package. Then, when the package is run within a virtual machine, you will be running the experiment exactly like it was run on the author's machine.

That can save a lot of pain.

[1] scikit-learn is a popular Python machine learning module.

Exactly.

What do you see as the future of tool-based reproducibility?

That's a good question. So, I can tell you what my dream is. My dream is that reproducibility will become a standard component of all computational environments. You should be able to work, do everything as you currently do, and with the click of a button, you will be able to retrieve everything that you did with essentially, zero additional work. This is what we should aim for. There has been substantial progress in the past few years, and nowadays, attaining reproducibility is much easier. There are lots of open-source tools, virtualization technology, clouds. But there are also gaps which can make the creation of reproducible results difficult in some scenarios. We need to better understand these gaps, and address a number of research and engineering challenges. I have been working towards convincing funders to have Programs to fill these gaps so that we can have reproducibility everywhere.

So, we might expect to see new calls for proposals that target those gaps?

If I am successful, yes.

> ***Let's get reproducibility to the masses without having to put any burden on them.***

You like to work on data management issues for emerging applications. What's the next big thing for the data research community in terms of applications?

There is a broad area of trust in the data and computation that I think is extremely important and has great potential for practical impact. And this ties back to what I mentioned that data and computation now are at the center of everything -- this sounds like a cliché, but it's actually true. As we have more and more people using computing and data, we need to have better mechanisms to guide them and help them build trust in what they do. We need to have better support for identifying issues, bugs in data, the computational steps executed, and in the computational environment – all of these can actually impact your results. This is a huge area with lots of very interesting research problems, and there is a huge unmet need for this right now.

Great, sounds very interesting.

There has been growing interest in machine learning models. You have your machine learning pipelines, and you want to explain the results for those pipelines. I think that we should be asking a broader question, in addition to machine learning, we should seek to understand and explain computations in general – Machine learning is just one component of the data science pipeline. How you obtain the data, what you do with the data, the kind of preprocessing, computations all contribute to the results produced by machine learning tools.

You have been the chair of ACM SIGMOD for almost four years now. What changes have taken place during that time?

I've actually been looking at some of the plans from four years ago, what I had in mind when I became chair – a retrospective look at what I wanted to do and what I actually did. One of the challenges that I identified is the fact that our community is growing and it's becoming more and more diverse. When I say diverse, I mean in all different aspects – not just demographics, but also in research areas. The status quo is that papers have to be about specific, traditional topics, for example, database engine. There is also a mindset for what a SIGMOD paper looks like. One goal that I had was to open this up. We are a big community – how can we actually let all flowers bloom? And how can we recognize all the different types of work? Our goal as researchers is to have impact and to have impact, we need to work on many different problems.

There have been a number of changes at SIGMOD that go in that direction. We have a new Applications track that aims to bring people from different areas to work with us, with our community, that was introduced by Divesh and Stratos and is now being refined by Amr and Angela. There has also been a lot of work by the PC chairs of SIGMOD to educate the reviewers to recognize different types of work and also review papers with a positive mindset, what AnHai and Wang Chiew termed as "review to accept". This is a step towards changing the culture that "we want these kinds of papers, and if a submission deviates, it is not worthy of SIGMOD." This requires educating reviewers to try and recognize novelty in different types of work, and contributions that will not only move our community forward but also lead to impact.

Do you have any words of advice for fledgling or mid-career database researchers?

Choose the right problem to work on. Selecting a problem that matters and has potential for practical impact is very important (at least to me). And not only

that, choose something that you are passionate about because things are hard, and it is a lot easier when you are passionate about something to actually keep on it even when you fail over and over again.

Amid all your past research, do you have a favorite piece of work?

It depends, Among my past projects, the body of work that we did on provenance and VisTrails is probably my favorite because it addressed an end-to-end problem, it involved theoretical and practical research, interdisciplinary collaborations. We went from the conception of the initial idea to doing Computer Science research, applying this research to different scientific domains, developing and deploying software. The work that I am doing now on building trust, debugging and explaining computations is something that I am very passionate about. It is at a very early stage, but it is a good candidate to become a favorite.

If you magically had enough extra time at work to do one more thing, what would it be?

I would spend more time working towards mentoring young minority students. I am Latina, and there are very few of us there are in academia or in top positions in Computer Science and Data Science. So, I wish I had more time to devote to increase the representation of minorities in Computer Science. I am making some time for this in the summer. NYU Tandon has a program called ARISE[2] that recruits high school students from underprivileged communities, and they spend a month at NYU. My lab will host two ARISE students. I hope to devote more time to this and similar initiatives in the future.

If you could change one thing about yourself as a Computer Science researcher, what would it be?

This is a tough question. Career-wise, I think that if I look back, I would probably have tried to plan more and be more strategic – things happened, and I just did it. Maybe my life would have been easier had I planned, but maybe it would have turned out differently, and I am happy as is.

Thank you very much for talking to me today.

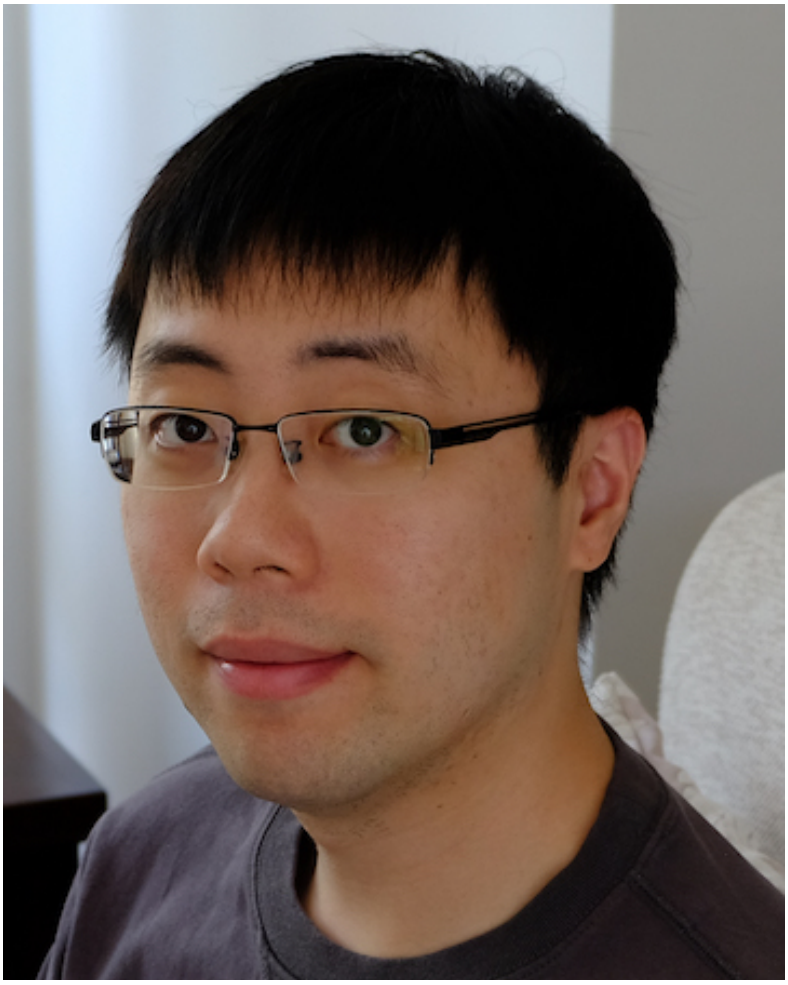
Thank you so much, Marianne. Nice talking to you.

You're welcome.

[2] https://wp.nyu.edu/k12/arise

Huanchen Zhang Speaks Out on Memory-Efficient Search Trees

Marianne Winslett and Vanessa Braganholo



Huanchen Zhang

https://people.iiis.tsinghua.edu.cn/~huanchen/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today I have here with me Huanchen Zhang, who is the 2021 winner of the ACM SIGMOD Jim Gray Dissertation Award for his thesis entitled Memory-Efficient Search Trees for Database Management Systems. After a postdoc at Snowflake, Huanchen is now an Assistant professor at Tsinghua University. His PhD is from Carnegie Mellon University, where he worked with David Andersen and Andy Pavlo. So, Huanchen, welcome!

Thank you.

What was your dissertation about?

My dissertation was focused on how to reduce the size of in-memory search trees in a database without compromising query performance. In many cases, the techniques introduced improve the end-to-end system performance in addition to the memory saving. We observed this problem back in 2015 when running the TPC-C benchmark on H-Store, which is a main memory OLTP database. And we found in the experiment that the indexes, which are B+ trees, eat up more than half of the database memory.

That's insane, right? The indexes are literally larger than the actual data you store. And someone may argue that the DRAM is getting cheap, so just don't worry about it, but that's not true. If you compare the per-gigabyte cost between DRAM and SSD for the past seven, eight years, the price gap between them is increasing. So, the truth is that DRAM is getting relatively more expensive when compared to storage. Also, because of the rapidly growing database sizes, memory as a resource is actually even more precious than before.

[...] stay healthy. No research is more important than your health. So, get enough sleep, take a vacation when you feel stressed. Life is much more than just research. Enjoy it.

So, under this background, reducing the memory footprint of those search tree indexes makes a lot of sense. It improves the memory efficiency of database systems -- and a better memory efficiency eventually translates to better performance and a smaller bill.

When I talk about compressing indexes, people usually think of using block compression algorithms such as snappy, LZ4, or Zstandard at a page level. This approach works really well for disk-based indexes because the IO latency hides everything. But in the case of in-memory search trees, where we are talking about several million index operations per second, the decompression overhead of these algorithms is just too expensive.

So, my entire thesis is about designing new search tree data structures that are super-small in size and super-fast at the same time. I provided a three-step recipe in my thesis to achieve that goal. As the first step, we focused on static data structures because they are much easier to compress. What we do here is that we borrow the concept of succinct data structures from the algorithms community and sort of reinvent this technique from a systems engineering perspective so that it's fast enough to be used in practice in addition to being succinct. A representative data structure we built is called SuRF. SuRF is a range filter. It's like a bloom filter, but it can also handle range queries. And it's quite useful. It's been used in LSM trees today. Under the cover, it's a static trie data structure that we designed with a space consumption close to the optimal -- optimal meaning the minimum bits required by the information theory. And we managed to engineer it to be really fast, with a performance comparable to some of the fastest trees out there.

Once we have this fast and close-to-optimal-space, static data structure, the next step in our recipe is to relax this constraint of being read-only. Here we introduced the hybrid index design as an efficient way to handle individual inserts, updates, and deletes to the static tree, but with an amortized cost. The key idea is very simple. We put a dynamic tree as a write buffer in front of the static one, and we do periodic merges between them. With a clever merge strategy, we managed to bound the overhead of this extra merge step to be very small.

Now, the last step in my recipe deals with the index keys because the first two steps already pushed the structural overhead of a search tree to the minimum. So, the dominating factor in terms of space shifts towards the index keys. Now, there are several challenges of compressing index keys. The first is that the encoding has to be order-preserving, otherwise you won't be able to perform range queries on the tree. And the second is that you don't know all the keys beforehand: a user can insert arbitrary keys. So, the traditional dictionary encoding doesn't work here. These are the challenges we solved in the last piece in my dissertation. In one sentence, we've built a super-fast compression tool called HOPE that can encode arbitrary input strings while preserving their original order.

Altogether, these three steps form a practical recipe for achieving memory efficiency in search trees, and that's my thesis.

Have you seen impact from your thesis in industry?

Yes. The SuRF range filter is being used by several major internet companies in their LSM-tree engines.

Do you have any words of advice for today's graduate students?

Yeah, I do have many because my Ph.D. wasn't that smooth. But according to my observation, the main reasons for Ph.D. students to drop out are: (i) they feel they have accomplished nothing, so they lost confidence; and (ii) they have to give up because of bad health (either physical or mental health). So, my advice would be first, whatever project you're currently doing, finish it. Even if you think it's a dead-end, just wrap it up, publish the results somewhere, maybe on archive, before thinking about what's next. It's very dangerous if you just jump between topics and projects and end up not completing any of them -- it will destroy your confidence in getting things done. And my second advice obviously is to stay healthy. No research is more important than your health. So, get enough sleep, take a vacation when you feel stressed. Life is much more than just research. Enjoy it.

Thank you very much for talking with me today.

Thank you for having me.

VLDB 2021: Designing a Hybrid Conference

Philippe Bonnet
IT University of Copenhagen
phbo@itu.dk

Xin Luna Dong
Facebook
lunadong@fb.com

Felix Naumann
Hasso Plattner Institute, University of Potsdam
felix.naumann@hpi.de

Pınar Tözün
IT University of Copenhagen
pito@itu.dk

ABSTRACT

The 47th International Conference on Very Large Databases (VLDB'21) was held on August 16-20, 2021 as a hybrid conference. It attracted 180 in-person attendees in Copenhagen and 840 remote attendees. In this paper, we describe our key decisions as general chairs and program committee chairs and share the lessons we learned.

1. KEY DECISIONS

Our main goal when organizing VLDB 2021 was to foster high-quality interactions. We worked under the assumptions that there would be restrictions due to the Covid-19 pandemic, but that large indoors gatherings would be allowed, and international travel would be possible in August 2021. As a result, we designed the conference as an in-person conference with the possibility of remote attendance.

1.1 Program

To foster interactions, we prioritized in-person attendance (over fairness across time zones) and we encouraged live exchanges (over asynchronous communication).

We kept the traditional VLDB format, with plenary sessions and up to seven parallel sessions. We scheduled unique sessions for research papers, industrial papers, keynotes, tutorials, and workshops at a suitable local time in Copenhagen. We thus chose not to repeat sessions, as opposed to VLDB'20 and SIGMOD'21, to avoid diluting potential audiences. To be accommodating for remote attendees, both in Asia and in the Americas, we held plenary sessions in the afternoons in Copenhagen.

We encouraged speakers to give live talks, whether in-person or remotely. In addition, we collected prerecorded 10-min videos for all papers as backup and for archival purposes. Despite our encouragements, many remote attendees chose to use their recorded video rather than give a live presentation, despite their attendance of the session.

Poster and demo sessions were organized as purely virtual events to minimize cost and to ensure a safe setup for these traditionally close-proximity sessions during the pandemic. Finally, we included virtual-only roundtable sessions beyond the local daytimes due to the large number of remote attendees, and because of their popularity and effectiveness in previous virtual conferences.

While we encouraged workshop organizers to run in hybrid mode, we gave them the option to run virtual-only workshops. Six of the 13 workshops chose the virtual-only option.

1.2 Digital Platforms

The requirements for the digital platforms were to provide (i) a schedule for all attendees, (ii) access to live sessions for remote attendees, (iii) support for synchronous and asynchronous interactions among attendees and (iv) opportunities for sponsors to reach all attendees.

We wanted to minimize the total number of platforms, so we chose (1) *Whova* as the only entry point to the virtual part of the conference, and (2) *Zoom* to stream the sessions to remote attendees. We used YouTube and Bilibili for the prerecorded videos.

In Whova, we enabled the *exhibitor center* and the *artifact center*. The exhibitor center allows sponsors to customize their interactions with the attendees, while the artifact center allows paper authors to continue discussions beyond the sessions with other attendees.

In Zoom, we opted out of the webinar mode to further increase interactions among the attendees. While the webinar mode of Zoom is more secure against disruptive attendees, it creates an isolating

experience for both the attendees and the presenters. The conference organizers have the power to react to disruptions, rather than being pessimistic and proactively avoiding them, especially when conference access requires a paid registration.

We relied on Gateway for the setup and management of Zoom sessions. Gateway knows our conferences and the virtual platforms, such as Zoom, Whova, and YouTube, very well at this point, and their services were extremely valuable during the conference. We relied on the audio/video (A/V) equipment, network capacity and technicians from the venue to stream up to seven sessions in parallel. For in-person presentations, slides were streamed directly from the presenter's laptop (connected to Zoom), while audio and video was streamed from the venue's A/V equipment.

We introduced two new functions in the conference organization, digital platform chair and artifact chair, to manage consistency across the digital platforms and to guarantee the quality and the timely delivery of the pre-recorded videos, respectively.

The role of the *digital platform chair* has already become part of our conferences with the virtual format. This role is necessary to manage the content on the virtual event platform and coordinate with other parties about population of and updates on this content. Digital platform chairs are also the first responders when the conference attendees have questions about the virtual platforms of the conference.

The role of the *artifact chair* is essential to manage the process for collecting all the conference artifacts, such as pre-recorded videos, posters, etc., and coordinating all the parties that are involved from paper authors to Gateway (in VLDB 2021). During the process of archiving these artifacts, the artifact chair hands-off the necessary information and consents to the person responsible from archival.

In addition, as usual across all conferences, we also used the conference website to make the conference schedule information public and social media to promote the conference.

1.3 Cost and Fee Structure

The main cost of the conference is associated to the venue, catering, A/V equipment, and personnel as well as social events. The fixed cost of a hybrid conference is much higher than for a traditional conference because of the cost of A/V equipment and personnel needed to cover seven parallel sessions.

We used a professional conference organizer, Kuoni, to take care of interactions with the venue and all service providers as well as sponsors. Kuoni also took care of the registration site. We introduced a flexible registration process that allowed changing remote to in-person registrations and vice-versa, up to two weeks before the conference start date. In this way, attendees could accommodate changes in personal circumstances as well as changes in company or government policies during the pandemic.

Note that we complemented the professional service from Kuoni with contributions from student volunteers. In particular, we needed volunteers to periodically transfer the list of registered attendees from Kuoni to Whova.

We decided to introduce a significant fee for remote attendance in order to cover the costs associated with professionally live-streaming sessions from the conference venue. Our rationale was that these costs should not be shouldered only by authors, since they are among the ones contributing to the attractive content of the conference.

In addition and as usual, we minimized the cost for student attendees. Fellowships from the VLDB SPEND committee as well as NSF covered registration fees for 75 students attending remotely and 14 attending in-person.

2. LESSONS LEARNED

2.1 Session Recordings

We initially considered session recordings as optional and not part of our core requirements. This was a mistake: recording sessions and making them available to conference attendees should be a requirement for any hybrid conference. It brings a lot of value to remote attendees and in-person attendees alike, specially in a multi-track conference such as VLDB.

There was a high demand during the conference for the session recordings to be available as soon as a session finished. We decided to record all the sessions, unless session chairs, presenters or attendees rejected, and make them available to the attendees through Whova. However, it takes time and manpower to edit, render, and upload session recordings to YouTube. As a result, most sessions recordings were available only two to three days after the sessions. We decided to make these session recordings available to attendees in Whova for a while longer after the conference ended. Note that there were about ten registrations after the conference was over, showing the value and the necessity of session recordings for attendees.

2.2 Sources of Complexity

Several sources of complexity that are unique to a hybrid conference required more attention than expected.

2.2.1 Session planning

With remote presenters joining from many different timezones, a key design question was whether to organize the paper sessions primarily based on the topic of the papers or the timezone and attendance mode of the speakers. We chose to group the papers based on topic first to allow a more natural flow in each session. Then, we attempted to schedule each topic-based paper session at a timeslot that is the most ideal for the majority of the remote speakers in that session, based on their timezone, while avoiding topically similar sessions running in parallel.

2.2.2 Streaming sessions

The requirement to stream sessions from the venue as well as remote presenters is a source of complexity before and during the conference. Indeed, both the Zoom manager and the A/V technicians at the venue must know the presentation form for each presentation (live in person, live on Zoom, or pre-recorded video). Collecting this information from all authors before the conference is complex as it involves several tracks managed by different chairs (research, industry, tutorials, keynotes, workshops). This information must be consolidated and shared with both the Zoom manager and the A/V technicians in a format that is convenient for them (e.g., grouped by day, session time for Zoom managers and by day, room, session time for A/V technicians).

To eliminate sources of inconsistencies, we chose to minimize the number of persons in the organizing committee interacting with Zoom manager and A/V technicians. This resulted in less autonomy for workshop chairs, who needed to interact with the conference general chairs to prepare for and manage session streaming.

2.2.3 Session chairing

A hybrid session setup increases the responsibilities of session chairs. First, they have to give directions to speakers and attendees about the hybrid setup, such as informing them about where to stand with respect to cameras and how to speak to the microphone to be audible and visible. Then, they shall monitor both in-person and virtual attendees to prevent people from being disruptive for the session. Finally, they must bridge the in-person and virtual parts of the session by coordinating speakers and questions on both sides.

To deal with these increased set of responsibilities, we assigned two session chairs per session – one playing the traditional role of a session chair, the other acting as a stand-in for the online-participants, monitoring their questions. With the smaller number of senior researchers attending in person, we were forced to ask them to chair two or even three sessions, and if possible recruit an ad-hoc second session chair before the session began.

2.2.4 Enforcing consistency

The schedule for a traditional conference program mainly contains information about which paper is presented or who presents at each session. This information is also enough for the attendees to decide which sessions to attend. However, to be able to run the sessions of a hybrid conference, a schedule document must be created that contains the additional information on the Zoom links, video location information for pre-recorded videos, the presentation modes for each talk, etc. This requires coordination of information from several independent, globally distributed parties, and we introduced a scheduling chair to oversee this process.

2.2.5 Publishing session recordings

We received many requests to publish recordings of keynotes or individual presentations. Publishing recordings introduces legal issues linked to personal data. These issues should be clarified before the conference starts so that legal forms are available at conference registration time.

Session recordings tend to have several attendees appear in the recording for brief moments of time in addition to the speakers. Given GDPR, a process must be established to handle cases, where attendees repeal their consent afterward to either remove the corresponding videos or edit out the corresponding person.

The pre-recorded videos of papers will be archived on the PVLDB website, since they do not pose the same level of complexity for GDPR. Indeed, consent from the speakers is enough to archive them in a GDPR-compliant way.

2.3 Trade-offs

There is a fundamental trade-off between interactions and inclusiveness. Both VLDB'20 and SIGMOD'21 implemented a 24-hour format, which scheduled each session twice, allowing attendees from all timezones to catch all the sessions at a reasonable waking time. As for SIGMOD'20, we chose to not repeat sessions and thus ensure a larger number of

attendees per session.

Overall, interactions among in-person attendees and interactions between in-person and remote attendees were very fruitful. However, the setups we had on Whova to boost interactions across all attendees, such as exhibitor and artifact centers, were not as highly used as we envisioned. In the future, making an additional effort with more networking sessions or deploying an additional platform, such as Slack, to boost such interactions may be necessary.

The demo and poster sessions, held virtually, attracted almost only authors. We initially planned a Zoom breakout room per demo and poster. However, we turned poster sessions into ad-hoc roundtables to increase the quality of interactions among poster authors.

Roundtable sessions have become a very popular part of VLDB and SIGMOD in the past couple of years, leading to many fruitful discussions. For VLDB 2021, we had many very exciting roundtable topics lined up thanks to our dedicated roundtable chair, and chose to run these roundtables in parallel (four to seven at a time). However, this parallelism hurt the attendance of the roundtables. For future conferences, having not more than a couple roundtable sessions in parallel should be a design principle to increase attendance and interactivity.

2.4 Scale

With 180 attendees in Copenhagen, VLDB 2021 felt like a small, intimate conference, with a mix of senior researchers and students. Our choice of fee structure, made the conference financially viable, even with much fewer in-person attendees than we originally planned for. The model of hybrid conference we worked with should scale to larger number of in-person attendees. Whether this model would work with fewer in-person attendees remains an open question.

2.5 Sustainability

While the topic of sustainability may seem orthogonal to the hybrid conference design, the hybrid format has great potential to facilitate more sustainable conferences. Allowing people to attend a conference remotely allows cutting down the cost of (flight) travel in addition to making the conference more inclusive and accessible. Similarly, being flexible with workshop program and allowing some workshops to be virtual could facilitate more sustainable options for the future. We are still investigating, with the sustainability chair, whether there are good options to actively offset the estimated carbon footprint of the conference.

3. CONCLUSIONS

This paper summarized our design and the lessons we learned about the hybrid format of VLDB 2021. Hybrid conferences foster interactions and bring the community together in-person, yet allow people who are unable to travel still be part of the conference. We believe that the hybrid format for scientific conferences is here to stay and opens up new opportunities for everyone.

Acknowledgements

The authors want to thank the VLDB Endowment and the many, many colleagues contributing to the success of VLDB 2021! Without the very high commitment of volunteers in our community, major conferences like VLDB would be infeasible.