How Large Language Models Will Disrupt Data Management

Raul Castro Fernandez
The University of Chicago
raulf@uchicago.edu

Aaron J. Elmore
The University of Chicago
aelmore@uchicago.edu

Michael J. Franklin
The University of Chicago
mjfranklin@uchicago.edu

Sanjay Krishnan
The University of Chicago
skr@uchicago.edu

Chenhao Tan
The University of Chicago
chenhao@uchicago.edu

ABSTRACT

Large language models (LLMs), such as GPT-4, are revolutionizing software's ability to understand, process, and synthesize language. The authors of this paper believe that this advance in technology is significant enough to prompt introspection in the data management community, similar to previous technological disruptions such as the advents of the world wide web, cloud computing, and statistical machine learning. We argue that the disruptive influence that LLMs will have on data management will come from two angles. (1) A number of hard database problems, namely, entity resolution, schema matching, data discovery, and query synthesis, hit a ceiling of automation because the system does not fully understand the semantics of the underlying data. Based on large training corpora of natural language, structured data, and code, LLMs have an unprecedented ability to ground database tuples, schemas, and queries in real-world concepts. We will provide examples of how LLMs may completely change our approaches to these problems. (2) LLMs blur the line between predictive models and information retrieval systems with their ability to answer questions. We will present examples showing how large databases and information retrieval systems have complementary functionality.

PVLDB Reference Format:
Raul Castro Fernandez, Aaron J. Elmore, Michael J. Franklin, Sanjay Krishnan, and Chenhao Tan. How Large Language Models Will Disrupt Data Management. PVLDB, 16(11): 3302 - 3309, 2023.
doi:10.14778/3611479.3611527

1 INTRODUCTION

The recent technological advances in Large Language Models (LLMs), such as OpenAI's GPT models [10, 55], Google's Bard, and many others on the horizon, are capturing the imagination of industry, academia, and the general public. These models have an unprecedented ability to understand and synthesize complex fragments of both programming and natural languages. We believe that this advance in technology is significant enough to prompt introspection in the data management community, similar to previous technological disruptions such as the advents of the world wide web, cloud computing, and statistical machine learning. Simply put, LLMs challenge the *raison d'être* of the data management community: effective and accurate question answering over data.

This moment in time was theorized decades ago. Databases and artificial intelligence share a common research history due to the natural connections between database query answering and symbolic reasoning [35]. The marriage of these two fields has resulted in a number of results spanning from knowledge bases [74, 75], large-scale graphical modeling [27, 28], and probabilistic query answering [11, 54, 60, 61]. Large databases storing an enormous corpus of facts were always seen as the conduit for powerful and effective AI applications. In a sense, LLMs are a refinement of this vision but crucially obviate the need for explicit data structuring or schema design. They are trained on large corpora of unstructured data and discover structure through statistical learning. This approach has been more effective than many researchers would have expected, and prototypes such as ChatGPT can answer complex questions with a reasonable degree of accuracy. As such, the rise of LLMs is the most significant shift in the relationship between databases and artificial intelligence that we have witnessed so far.

The widespread adoption of LLMs will disrupt many core data management problems. With their advanced natural language processing capabilities, LLMs will enable organizations to extract valuable insights from unstructured data sources, such as text documents and social media posts. This, in turn, will drive a more efficient and effective use of data, leading to better decision-making and business outcomes. The impact of LLMs on data management will be far-reaching and lead to new industry standards and best practices. This paper presents three key takeaway messages.

1. LLMs represent a fundamental shift in NLP capabilities. In the last 10 years, we have moved from word2vec models [51], which could answer simple natural language analogies, to models that can synthesize both text and code from arbitrary natural language prompts (with reasonable accuracy). This new synthesis capability leads to a number of intriguing data management questions around data governance, factual accuracy, and incorporating new data.

2. LLMs will force us to revisit key data management problems. In addition to these new opportunities, LLMs will affect traditional data management problems as well. A number of hard database problems, namely, entity resolution, schema matching, data discovery, and query synthesis, hit a ceiling of automation because the system does not fully understand the semantics of the underlying data.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 16, No. 11 ISSN 2150-8097.
doi:10.14778/3611479.3611527

Author list in alphabetical order.

3. LLMs Will Get More Efficient. A widespread concern about LLMs is their immense training and inference costs. In our opinion, these costs are over-emphasized as limitations. When AlexNet was first introduced in 2012, it took six days to train [42] and required specialized software. In 2018, a far more accurate Resnet-50 model can be trained in 18 minutes on AWS [34] (a 480x time reduction in 6 years). We believe the same hardware, software, and model efficiency trends will occur in LLMs; though the rate of improvement is unknown. Furthermore, the proliferation of fine-tuning solutions indicates that training an entire LLM from scratch will be rare.

When we initially submitted this manuscript in March 2023, we posited that such a strong statement about the future of LLMs might be "premature". We simply did not anticipate how much the space would change in the subsequent four months. There are now numerous commercial, open-source, and academic projects addressing the very problems discussed in the vision paper [5]. In reflection, these emerging systems do not change our vision, but actually emphasize the need for deep and rigorous research by the data management community into how LLMs should affect systems and practice. Individual researchers in our community have long contributed to the problems we discuss in this paper (see related work in Section 2.3). Furthermore, we will use this paper to preserve the unique snapshot in time when it was published to compare and contrast the authors' perspectives at submission versus publication. We analyze the connection of LLMs and databases in Section 2. We follow with a discussion of problems that will be disrupted in Section 3 and opportunities that will arise in Section 4 before offering conclusions in Section 5.

2 A SHIFT IN NLP AND DATA MANAGEMENT

In this section, we discuss the relationship between LLMs and data management and overview LLMs from different perspectives.

2.1 Large Language Models (LLMs)

By LLM we refer to a class of models that synthesize long textual outputs in response to a prompt. Today's LLMs share these traits: (1) they are trained over a large corpus of web data that includes natural language and code, (2) they are architecturally based on transformer networks [70], and (3) they have significantly more parameters than traditional ML models. Specifically, we concentrate on the GPT line of models [10], including ChatGPT, and smaller variants such as LLAMA [62] or ALPACA [2], which are widely available as of May 2023. While the community does not fully understand why they are so effective, LLMs are still a major breakthrough in information retrieval and are already integrated in search engines [3], efforts to understand how LLMs work continue growing [1, 16, 17, 24]. While the details of these models might evolve, the new functionality that they bring (i.e., textual synthesis in response to a prompt) is a significant enough change to force us to think about their relationship with data management systems.

Is an LLM a database? What is the boundary, if any? We can approach this question from different angles:

An LLM is an Information Retrieval Tool. Users have an information need, and they articulate it through various means, such as SQL queries, keywords, or natural language questions. The goal is for the system to deliver relevant information without the user worrying about the intricacies of how it is retrieved. From the user's perspective, the underlying technologies, whether it be Google, databases, the yellow pages, or LLMs, are all viewed as offering a similar abstraction. There is a query and there is a response.

In this sense, an LLM is simply the ultimate declarative abstraction for an information retrieval tool. The "queries" are posed as natural language prompts and information is "retrieved" accordingly. The definitions of query and retrieval are imprecise here. The nuance is that LLMs are able to answer queries for which the data are not in the training set through extrapolation. However, the inner workings of LLMs, specifically the principles of large neural networks and attention mechanisms, are not as well understood and LLMs have well-known failure modes with factual inaccuracies and hallucinations [46, 49, 79].

An LLM is a Synthesis Tool. Alternatively, some use LLMs as tool for code or text synthesis. For example, a data engineer might ask "write a Python regular expression that extracts data within curly braces". Or, a data governance administrator may ask "write a draft privacy policy for a website that collects profile pictures from user's webcams." In both cases, the LLM produces an initial draft that the engineer or the administrator can iterate on.

However, unlike recent tooling in "coding assistants" [7, 53], it is not purpose-built for programming. The same model might be able to solve a variety of programming language as well as natural language synthesis tasks. For example, one might be able to translate an english description of an application into a relational data model.

An LLM is a Data Enrichment Tool. The closed world assumption has historically underpinned relational databases, namely, facts not in the database are considered irrelevant to the query at hand [58]. As the size and scope of organizational data have grown, more organizations have understood that knowledge from outside the relational database, either from external corpora or unstructured data, is valuable to augment structured data. This idea has manifested itself in a number of different research projects, including work on data/knowledge integration [22, 23], knowledge bases [18], web data extraction [12, 25, 48], semantic web [6], and even database parameter tuning [63]. Such work has provided crucial "open-world" context to organizational data such as rich type inference, language models for integration, and the relationships between the database and real-world entities.

Each subsequent generation of information extraction tools has taken further advantage of data in the open world to enrich a database. It is here that LLMs are a technological discontinuity. LLMs can increasingly generate human-like language and perform a wide range of natural language processing tasks, such as text classification, text generation, and machine translation. The most advanced LLMs (GPT-4 as of May 2023) consist of billions of parameters trained on large volumes of data. This extensive training enables LLMs to capture a vast amount of knowledge and contextual understanding of the language they were trained on. As a result, they can perform complex language tasks with remarkable accuracy, consistency, and coherency. However, the softer notion of information structuring leads to known issues with incorporating new data accurately.

2.2 Critique

None of these analogies for LLMs are quite accurate. Simply put, LLMs are a different technology. Classical data management systems represent data through symbols (via different data types) that are associated with their meaning, and relationships between those symbols that are often captured through specific layouts on disk, along with metadata describing the structure of the data. In contrast, LLMs represent data in a fundamentally different way. LLMs encode text data as high-dimensional vectors of floating-point numbers, with each dimension capturing a particular semantic feature of the text. The resulting representation is stored in a multi-layered neural network trained to perform specific tasks. Query evaluation in LLMs involves performing algebraic operations on the numerical representations of the data rather than searching for specific symbols.

On the one hand, this method of structuring information leads to interesting integrative and extrapolative effects. LLMs can synthesize content not exactly seen in their training set [41]. They can also combine content in the training set in novel and surprising ways [73]. On the other hand, they are highly complex and unpredictable tools, e.g., since the paper's submission initial evaluations suggest behavior changes and potential regressions in GPT-4 [15].

While the technology is still evolving, a few aspects of it are clear right now: (1) LLMs can provide a bridge between programming languages and natural language that have far more functionality than past efforts [40, 43, 67]; (2) LLMs consolidate intelligence – the same model/tooling is useful across the entire data lifecycle; (3) LLMs integrate data with real-world context. These three strengths of LLMs change the way that certain database problems can be approached, as we explore next.

2.3 Related Work

Researchers in our community have long been contributing to many of the questions we raise in this paper. They have considered how pretrained language models [20, 47] can be used to configure databases [65], to facilitate writing SQL queries [64], to understand how to represent tabular data in formats more amenable to use with the above models [13, 82] and to address problems in data management [45, 78]. And there is similarly a large and growing body of work on optimizations to training and inference time [50, 66, 80, 81].Even more recently, there are a few works that consider generative large language models [52] (the target of this paper, see Section 2.1) for data management problems. We expect that much of the previous work will also leverage generative LLMs.

3 HARD PROBLEMS BECOME EASIER

LLMs will change many areas of data management. In this section, we focus on problems that have been historically difficult to address and that we think will benefit the earliest from LLMs. What were once AI-complete problems will become trivial as we tap into AI-capable agents; tedious but easy problems will be automated away.

3.1 Data Integration is a problem of Long Tails

Data management has provided solid solutions for query optimization, query processing, storage management; in a varied set of scenarios including small devices, single-node deployments, and large clusters. But the success on these tasks has not translated to data integration and other "hard-to-formalize" problems that we consider core to data management and our community [21, 32]. Today, data integration roughly means having a solution to *schema matching and mapping*, *entity resolution*, *alignment*, *data quality*—which is its own world—and more. Each of these problems is intellectually interesting and the amount of work the community has poured into it, and the progress made, is tremendous. And yet, data integration remains a difficult challenge because of the variety of scenarios in which it manifests, and the lack of practical tools.

Variety of Scenarios. While a SQL query expresses precisely what data to retrieve, this degree of precision is difficult to obtain when one tries to integrate two datasets. What does that even mean? "Meaning" is where many issues are born: what the person who produced a dataset meant when they designed the schema is not necessarily aligned with what the consumer believes that schema means. Semantic ambiguity makes data integration hard.

Tails. There are easy integration cases. And there are hard cases. There has been much success in addressing the easy ones. An example is program synthesis [31] for data transformations [30]. But the hard integration cases, those in the tail, mean that the problem, as a whole, still requires tedious hand-holding by humans that must curate and drive the wrangling process. This is related to the famous "80% of effort analysts spend ...".

LLMs will help in two ways. First, they will offer a new tool to ameliorate the semantic ambiguity problem. Second, they will assist in providing code snippets to solve the many "easy" cases at the tail, thus reducing the need for hand-holding and increasing the cases where integration is automated. Even general-purpose, proof-of-concept LLMs such as ChatGPT do well in many tasks that would be hard to automate, consider the following examples:

Us. *"In a tabular dataset, I have a column named "Addresses" with two values. The first value is "1210, S Indiana Ave" and the second value is "3333, N Richmond St". In addition to this tabular dataset, I have another one, with values: "North Richmond, 3333" and "South Indiana Avenue". Could you help me align the values from the first with the values from the second and represent the values all with one single representation?"*

ChatGPT:

```
# Mapping dictionary
mapping =
    { '1210, S Indiana Ave': 'South Indiana Avenue',
    '3333, N Richmond St': 'North Richmond, 3333' }
```

Us. *"standardize this address: "1210, S Indiana Ave""*

ChatGPT:

```
1210 S Indiana Ave, City: Chicago, State: IL, Zip code: 60605
```

Note that the city, state, and zip code are assumed based on the address being "1210, S Indiana Ave". If the actual city, state, or zip code is different, they should be replaced accordingly.

When we asked ChatGPT to combine a table with first name and last name with one with full name it did equally well. When we asked it to provide a code snippet to check integrity constraints of a zip code in the US, it immediately offered a detailed program that was easy to plug in existing code. That these cases were solved with ***no*** effort suggests that with a bit of tuning, LLMs will be a boost to data integration. And the good news is that the community is already exploring this area, consider the early Ava [37] and a more recent vision paper on the potential of LLMs for data wrangling [52].

Still, LLMs are sometimes wrong, and, worse, subtly wrong. The latter case compounded with humans' confirmation biases pose a challenge. We must make LLMs more robust and identify ways of having them work with humans in a productive manner.

3.2 Democratizing Data Access

SQL was designed to resemble the English language [14]. The idea was to design a language accessible to most users (at least those who speak English). But what is better than a language that looks like English is English. So the community has invested much in natural language interfaces that take natural questions and translate them into SQL [36, 44, 64].

LLMs will solve this problem for most queries: it is conceivable that very complex queries will still require a human expert for a while. But easing access to the bulk of queries will truly democratize data processing. For example, when we asked ChatGPT to translate the following request into SQL:

Us. *Assume a schema with 2 tables:*

em(emp_no, name, birthday, gender, hire_date, department_name)
de(dept_name, name, manager)

Can you write a SQL query that tells me, for each employee, the manager of the department the employee works for?

ChatGPT:

```
SELECT e.name AS employee_name, d.manager AS department_manager
FROM em e
INNER JOIN de d ON e.department_name = d.dept_name;
```

The ability to interact with data management systems via natural language also challenges the standard practice of interacting with symbolic logic systems. For example, it would be useful if LLMs provided an explanation of their answer, so those querying it can verify its correctness. Indeed, ChatGPT gave a detailed justification for the answer to the query above, which, if you noticed, does not specify constraints, uses different naming, and refers to "em" and "de" as the 'employees' and 'department' tables:

> *...this assumes that the em table contains a foreign key department_name that references the dept_name field in the de table and that the de table contains a manager field with the emp_no of the manager. If this is not the case, you would need to modify the query to join the tables together using their respective keys [...] if you want to retrieve the manager's name instead of emp_no, you would need to join the em table again to get the manager's name based on emp_no.*

LLMs will permit more people than ever before query systems and will facilitate interpreting and verifying the results.

3.3 Data Discovery

Data discovery is the problem of identifying and retrieving data that satisfies an information need [26]. The problem is difficult for at least two reasons: i) the large volume of unstructured, semi-structured, and structured data available; ii) the difficulty in articulating the information need. LLMs have shown a remarkable capacity to retrieve (and formulate) relevant information, all stemming from their training data which consists of a non-trivial portion of the Internet. There will be rapid advances in adapting LLMs to the enterprise setting, given the promise LLMs have shown to date and the growing discovery needs. We envision a future where all data sources within an organization are part of a large enterprise-LLM that lets employees search for relevant information. Several challenges impair this vision:

Scalability. LLM's quality depends on data, compute resources, and model size [39]. Large organizations have data. But large models will remain difficult and expensive to train, vastly reducing the number of enterprises benefiting from such solutions. Smaller organizations will be in an even more dire situation without even having all relevant data at hand: they may need to form coalitions with other organizations in the same sector, as we discuss in Section 4.1. Even for large organizations that can afford building LLMs, vector sizes are huge and continue growing. Hardware and algorithmic advances will help move the needled in the right direction, but they will need to be aided by all the technical expertise developed over decades, companies, and interactions on data access methods, compression, and more.

Govern Data Flows. Building enterprise-LLMs requires controlling data flows, which remains an elusive problem. Despite the difficulty, the renewed interest and pressure to ripe the benefits of LLMs will bring in solutions. Those solutions must help maintain compliance, respect privacy, and permit other access control restrictions. This is extremely challenging when the provenance of the output generated by an LLM is not as easy to conceptualize as that of, say, a SQL query. We discuss this in more detail in the opportunities ahead (see Section 4.2).

There are already many approaches to attack this data discovery problem using tables from the Internet [8, 19, 77], and our group has made progress in addressing the challenge within enterprise scenarios [72], where collecting training data is difficult. We expect many rapid advances in the coming years.

3.4 Evolving System Support

Neural network architectures rely on large vector of vectors to store parameters, which are growing exponentially with LLMs (i.e. GPT-4 is rumored to have 100 trillion parameters, whereas GPT-3 has over 175 billion parameters). These large vectors are costly to store and query. Ideas from database systems should be explored to help exploit access patterns, storage size, and potential indexing. For example, prior database work has explored using columnar systems for debugging models via effective intermediate results [69]. Similar approaches could be used for effective versioning for incremental training or crash-recovery. Given the scale of the vectors (both in terms of quantity and size), moving them in and out of GPUs introduces a number of I/O concerns that will limit scalability; addressing the scalability challenge will require codesigning hardware and software. Consequently, moving vectors in and out of storage or across distributed nodes will provide many opportunities for the underlying storage systems. Another opportunity for database systems is to provide additional support for developers. For example, vector databases that optimize for sparse representations for approximate similarity search. Recent projects such as Pinecone [4], VAQ [56], and FAISS [38] explore database-centric approaches to offering high performance vector-search functionality.

3.5 A Bright Future, Not without Challenges

As we learn how to use LLMs to solve data management problems, we anticipate a host of new associated challenges:

ETL Nightmare. Briefly, ETL software moves data from A to B, possibly changing its format on-the-fly. Many pipelines are exceedingly complex to build because they must target every special case in the data, e.g., transforming dates no matter the format in which they appear. LLMs will make building ETL pipelines faster because they will help developers target these numerous tail events more easily. Lowering the barrier to building these ETL pipelines will result in an increase in the total number of ETL pipelines, thus raising maintenance costs. Reigning over the increasing complexity will require novel contributions.

Trust. All the potential problems of LLMs in general scenarios apply to data management as well. If an analyst obtains a SQL query from the LLM, can they trust the SQL query to perform the operation they wish? An increased ability to explain and justify answers will be needed across the board, and that includes data management as well. Techniques to memoize previously working answers, and to marry information retrieval with LLMs will surely help achieve this goal.

Updating LLMs. Today's LLMs are *static*. The current version of ChatGPT (May 24th, 2023) was trained with data from months ago. Incorporating new data into LLMs without expensive retraining is a challenge. Successfully tackling this challenge will require a combination of system and machine learning contributions.

Lack of data. While the open Internet is full of rich data sources, this is not always true in enterprises, where data may be in semi-structured and structured formats. Ensuring this data can be incorporated into LLMs will likely demand changes in the training procedures. However, today's LLMs are already doing a great job in producing structured programming language snippets, and the efforts in table representation [8, 77, 82] are a good indication that this challenge will be addressed soon.

4 NEW HORIZONS

LLMs will introduce new opportunities. In this section, we discuss a few lines that our group is working on and that we believe the data management community has much to contribute.

4.1 Incentivizing Data Sharing

Today, LLMs are built and operated by large companies. Soon, they will be accessible to many more. But they will remain quite expensive to train and to deploy. There are two scarce resources:

Collecting Domain-Specific Data. Organizations that want to train LLMs on their domain may not have sufficient data to train good models and would benefit from forming coalitions with others.

Provisioning Sufficient Compute. There is a tension between building smaller (cheaper) LLMs, and scaling them to achieve more capabilities. For the foreseeable future, training LLMs will remain expensive, posing a barrier for many organizations.

This introduces several opportunities. First, scarcity of data and compute will incentivize the creation of LLMs as a "public commons": models trained by governments and made available to their citizens, thus avoiding full dependence on private organizations. We already see efforts in this direction, including the non-profit LAION [59] that collects data from the crowd and builds open source models, and petitions by scientists in Europe and the US to get their governments to build open-source LLMs. Plus, online repositories of datasets and models such as HuggingFace [76] and OpenML [68] will continue to grow.

Second, we will see an increased need for the formation of data-sharing consortia, groups of organizations that collaborate with each other to address the problems introduced by scarce resources. Data-sharing consortia helps in two ways: i) participants pool their data, leading to access better training datasets; ii) participants *cost-share* the training and deployment of these models.

The data management community—together with collaborators in relevant disciplines such as economics, mechanism design, and human-computer interaction—has an opportunity to contribute to the design of platforms to incentivize and facilitate data sharing.

4.2 Who's data is it? Provenance in LLMs

Over the last few decades, large Internet companies such as social networks and search engines have leveraged individuals' data to create a profitable ad business. The numerous problems that stem from the indiscriminate collection of this information will soon propagate to LLMs, which are built off data available on the open Internet. New terminology such as "surveillance capitalism" [83], "data dividends" [71], "data-as-labor" [57] has appeared to describe the problem and potential solutions. No solution has taken ground yet, although there is a renewed interest triggered by the public becoming aware of the downsides of indiscriminate data collection.

A question of ownership. Who owns the data used to train LLMs? Are those who provide the data aware? do they consent? and if so, are they adequately compensated for their contribution? Today, the answers to these questions are negative, and without any intervention, they will remain so. Companies leveraging data without attribution will be a growing problem. If individuals from whom the value of these models stems (whether via the code they publish on GitHub, their blog posts, and other contributions) are dispossessed of their contribution, LLMs cannot grow in a profitable way for society. And if we lose track of what input data led to an LLM output, we won't reign in the incoming challenge of detecting and addressing non truthful and biased outputs, which may be weaponized as misinformation. A full solution to these problems is complex but keeping track of provenance must be part of such a solution.

Provenance as an enabling mechanism. The challenges of understanding the provenance of LLMs are daunting. At a minimum, we should be able to tell whether a data source was used as part of an LLM. More challenging, we must be able to associate output with the input data. We will then need to define a notion of "contribution" that measures how much is the output based on input sources. This is where the difference between model and database blurs. In databases, the attribution can be solved using provenance techniques [29] to determine precisely the connection between output and input. In models, the output is a distributed representation over all input, making it more challenging to establish a connection.

Value data as we value grain. Perhaps a more adequate model to attribute outputs to inputs is to relinquish the goal of fine-grain connection and instead concentrate on classifying input data into a discrete set of *grades*, each of which corresponds to a degree of data quality. This is inspired by how wheat farmers in the late 19th century proposed a *grading* system for their wheat, later passed by Congress on the Federal Grain Standards Act [9]. This grading system is widely recognized as a step forward in building a better wheat market, one that even led to the formation of "pools" (see Section 4.1) [33]. This grading system would standardize data, avoiding the need for fine-grained valuation, and transforming that challenging problem into a simpler classification of data into grades. Solving the economics of making such a mechanism work is an interesting problem. Data management must play a role in providing the tooling and infrastructure to implement such mechanisms.

On compensating contributors. And after a quality is assigned, the remaining problem is to adequately compensate contributors. Monetary compensations are just one type. But many of the most successful compensations in CS are reputation-based, such as the reputation associated with contributing to the Linux Kernel or other influential open source projects.

An appropriate reaction to the challenges may establish *provenance* as a first-class citizen that is expected in a kind of new "social contract" i.e., consumers of LLMs outputs will demand knowing where their data came from and how it came to be, thus putting upstream pressure on the systems used to generate such data. This would have positive repercussions way beyond the use of LLMs.

4.3 Data Governance via Data Markets

LLMs are revitalizing a long-running discussion on the governance of data, including questions of privacy, attribution, and value. For example, OpenAI has been criticized for keeping the LLMs they train closed without giving attribution to the otherwise often public sources of data used to train the models. Data governance is not a traditional data management problem but it must become one as it will continue growing in importance with the development of LLMs and the continued reliance on data. Our vision is that *data markets* understood as *environments where agents exchange data* are an approach to address data governance as we explain next.

Applying this "data market lens" helps articulate problems of data governance, control of data flows, attribution, liability, and value extraction. For example, if we apply this lens to the problem of incentivizing data sharing (Section 4.1) one can immediately recognize the agents as the participating organizations, the exchange as the pooling of these resources, and data as the main resource exchanged to build and operate LLMs. And, when applied to the problem of ownership (Section 4.2), the lens concentrates on the agents' incentives and preferences. In that market, agents contribute to producing a model that will then affect other agents. We want to ensure the incentives and preferences of agents are honored according to a standard of social welfare. What this standard should be is up to the environment where the market operates. It could be a self-organized body of organizations or individuals. Or it could be a government. Whatever it is, we have no way of enforcing that today because these ideas are growing organically (e.g., pushed by organizations that naturally seek to further their own agenda). This growth is good news for innovation only if paired with appropriate responsibility towards affected citizens.

5 CONCLUSION: WHAT DOES CHATGPT "THINK"?

Us: *I would like to know your opinion on whether LLMs will disrupt data management, and how can we make sure the combined used of LLMs and databases will be productive.*

ChatGPT: *As an AI language model, I don't have personal opinions, but I can provide you with some insights and perspectives on the topic.*

LLMs have the potential to impact data management by enabling users to interact with data using natural language. This can reduce the need for technical expertise and make data more accessible to a wider range of users. However, LLMs also introduce new challenges, such as ensuring the quality of the generated output, and addressing ethical and legal concerns regarding privacy and ownership.

To ensure the productive use of LLMs in combination with databases, it is important to establish guidelines for their integration. This includes developing best practices for data preparation, model training, and output validation, as well as addressing issues related to data privacy, security, and governance. It is also important to invest in education and training to ensure that users understand the capabilities and limitations of LLMs, and can effectively use them in conjunction with traditional database management tools and techniques.

In summary, while LLMs have the potential to disrupt data management, their productive use requires careful consideration of both the opportunities and challenges they present, and the development of appropriate guidelines and standards to ensure their effective integration with traditional database management techniques.

Us: *When considering how LLMs will affect data management, give me a brief worst case scenario and a brief best case scenario.*

ChatGPT: *Sure, here are brief worst case and best case scenarios regarding the impact of LLMs on data management:*

Worst case scenario: LLMs become the primary method for querying and analyzing data, leading to a decline in the use of traditional databases and causing significant disruption in the data management industry. This could lead to a loss of jobs for professionals in the database management field and a lack of trust in the accuracy and security of LLM-generated insights.

Best case scenario: LLMs are integrated with traditional databases to improve data management capabilities, leading to more efficient and accurate querying and analysis of data. The use of LLMs allows for more intuitive and user-friendly interfaces, making data management more accessible to a wider range of users. This could lead to greater insights and discoveries from data and more innovative uses of LLMs in the data management industry.

Final Word (Us): This computer generated discussion shows that LLMs mark an unprecedented improvement in natural language capabilities. We will leave the readers with a single question that summarizes the vision of this paper: Should Chat-GPT have been included as an author?

REFERENCES

[1] 2022. A slight-less-magical perspective into autoregressive language modeling: count, compress and prune. https://kyunghyuncho.me/talks/.

[2] 2023. Alpaca: A Strong, Replicable Instruction-Following Model. https://crfm.stanford.edu/2023/03/13/alpaca.html.

[3] 2023. Confirmed: the new Bing runs on OpenAI's GPT-4. https://blogs.bing.com/search/march_2023/Confirmed-the-new-Bing-runs-on-OpenAIâĂŹs-GPT-4.

[4] 2023. Pinecone: Vector Database for Vector Search. https://www.pinecone.io/.

[5] 2023. Survey: Massive Retooling Around Large Language Models Underway. https://www.forbes.com/sites/aparnadhinakaran/2023/04/26/survey-massive-retooling-around-large-language-models-underway/?sh=53bf152814a1.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american* 284, 5 (2001), 34–43.

[7] Pavol Bielik, Veselin Raychev, and Martin Vechev. 2015. Programming with" big code": Lessons, techniques and applications. In *1st Summit on Advances in Programming Languages (SNAPL 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[8] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2022. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[9] James E Boyle. 1925. The Farmers and the Grain Trade in the United States: An Interpretation of the Present Pooling Movement. *The Economic Journal* 35, 137 (1925), 11–25.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[11] Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *Proceedings of the VLDB Endowment* 2, 1 (2009), 1090–1101.

[12] Michael J Cafarella, Jayant Madhavan, and Alon Halevy. 2009. Web-scale extraction of structured data. *Acm Sigmod Record* 37, 4 (2009), 55–61.

[13] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2021. Embdi: generating embeddings for relational data integration. In *29th Italian Symposium on Advanced Database Systems (SEDB), Pizzo Calabro, Italy*.

[14] Donald D Chamberlin. 2012. Early history of SQL. *IEEE Annals of the History of Computing* 34, 4 (2012), 78–82.

[15] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's behavior changing over time? *arXiv preprint arXiv:2307.09009* (2023).

[16] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Florence, Italy, 276–286. https://doi.org/10.18653/v1/W19-4828

[17] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. 2020. On the Relationship between Self-Attention and Convolutional Layers. arXiv:1911.03584 [cs.LG]

[18] Christopher De Sa, Alex Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2016. Deepdive: Declarative knowledge base construction. *ACM SIGMOD Record* 45, 1 (2016), 60–67.

[19] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record* 51, 1 (2022), 33–40.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[21] AnHai Doan, Alon Halevy, and Zachary Ives. 2012. *Principles of data integration*. Elsevier.

[22] AnHai Doan and Alon Y Halevy. 2005. Semantic integration research in the database community: A brief survey. *AI magazine* 26, 1 (2005), 83–83.

[23] Xin Luna Dong. 2018. Challenges and innovations in building a product knowledge graph. In *Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining*. 2869–2869.

[24] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position Information in Transformers: An Overview. *Computational Linguistics* 48, 3 (09 2022), 733–763. https://doi.org/10.1162/coli_a_00445 arXiv:https://direct.mit.edu/coli/article-pdf/48/3/733/2040503/coli_a_00445.pdf

[25] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence* 165, 1 (2005), 91–134.

[26] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[27] Lise Getoor, Nir Friedman, Daphne Koller, Avi Pfeffer, and Benjamin Taskar. 2007. Probabilistic relational models. *Introduction to statistical relational learning* 8 (2007).

[28] Lise Getoor and Ben Taskar. 2007. *Introduction to statistical relational learning*. MIT press.

[29] Todd J Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 31–40.

[30] Sumit Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. *ACM Sigplan Notices* 46, 1 (2011), 317–330.

[31] Sumit Gulwani, Oleksandr Polozov, Rishabh Singh, et al. 2017. Program synthesis. *Foundations and Trends® in Programming Languages* 4, 1-2 (2017), 1–119.

[32] Alon Halevy, Anand Rajaraman, and Joann Ordille. 2006. Data integration: The teenage years. In *Proceedings of the 32nd international conference on Very large data bases*. 9–16.

[33] Lowell D Hill. 2021. Grain grades and standards. In *Grain Marketing*. CRC Press, 121–158.

[34] Jeremy Howard. 2018. Now anyone can train Imagenet in 18 minutes. https://www.fast.ai/posts/2018-08-10-fastai-diu-imagenet.html.

[35] Richard Hull and Roger King. 1987. Semantic database modeling: Survey, applications, and research issues. *ACM Computing Surveys (CSUR)* 19, 3 (1987), 201–260.

[36] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2073–2083.

[37] Rogers Jeffrey Leo John, Navneet Potti, and Jignesh M Patel. 2017. Ava: From Data to Insights Through Conversations.. In *CIDR*.

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

[39] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

[40] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23.

[41] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050* (2023).

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[43] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 563–569.

[44] Fei Li and Hosagrahar V Jagadish. 2014. NaLIR: an interactive natural language interface for querying relational databases. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 709–712.

[45] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584* (2020).

[46] Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848* (2023).

[47] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

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

[49] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of ACL*.

[50] Xupeng Miao, Yujie Wang, Youhe Jiang, Chunan Shi, Xiaonan Nie, Hailin Zhang, and Bin Cui. 2022. Galvatron: Efficient Transformer Training over Multiple GPUs Using Automatic Parallelism. *arXiv preprint arXiv:2211.13878* (2022).

[51] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[52] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *arXiv preprint arXiv:2205.09911* (2022).

[53] Nhan Nguyen and Sarah Nadi. 2022. An empirical evaluation of GitHub copilot's code suggestions. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 1–5.

[54] Feng Niu, Che Zhang, Christopher Ré, and Jude W Shavlik. 2012. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *VLDS* 12 (2012), 25–28.

[55] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[56] John Paparrizos, Ikraduya Edian, Chunwei Liu, Aaron J. Elmore, and Michael J. Franklin. 2022. Fast Adaptive Similarity Search through Variance-Aware Quantization. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 2969–2983. https://doi.org/10.1109/ICDE53745.2022.00268

[57] Eric A Posner and E Glen Weyl. 2018. Radical Markets. In *Radical Markets*. Princeton University Press.

[58] Raymond Reiter. 1981. On closed world data bases. In *Readings in artificial intelligence*. Elsevier, 119–140.

[59] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402* (2022).

[60] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. 2015. Incremental knowledge base construction using deepdive. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, Vol. 8. NIH Public Access, 1310.

[61] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. 2011. Probabilistic databases. *Synthesis lectures on data management* 3, 2 (2011), 1–180.

[62] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[63] Immanuel Trummer. 2021. The case for NLP-enhanced database tuning: towards tuning tools that" read the manual". *Proceedings of the VLDB Endowment* 14, 7 (2021), 1159–1165.

[64] Immanuel Trummer. 2022. CodexDB: Synthesizing code for query processing from natural language instructions using GPT-3 Codex. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2921–2928.

[65] Immanuel Trummer. 2022. DB-BERT: a Database Tuning Tool that" Reads the Manual". In *Proceedings of the 2022 International Conference on Management of Data*. 190–203.

[66] Taegeon Um, Byungsoo Oh, Byeongchan Seo, Minhyeok Kweun, Goeun Kim, and Woo-Yeon Lee. 2023. FastFlow: Accelerating Deep Learning Model Training with Smart Offloading of Input Data Pipeline. *Proceedings of the VLDB Endowment* 16, 5 (2023), 1086–1099.

[67] Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*. 1–7.

[68] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2014. OpenML. *ACM SIGKDD Explorations Newsletter* 15, 2 (jun 2014), 49–60. https://doi.org/10.1145/2641190.2641198

[69] Manasi Vartak, Joana M. F. da Trindade, Samuel Madden, and Matei Zaharia. 2018. MISTIQUE: A System to Store and Query Model Intermediates for Model Diagnosis. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (Eds.). ACM, 1285–1300. https://doi.org/10.1145/3183713.3196934

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[71] Nicholas Vincent, Yichun Li, Renee Zha, and Brent Hecht. 2019. Mapping the Potential and Pitfalls of" Data Dividends" as a Means of Sharing the Profits of Artificial Intelligence. *arXiv preprint arXiv:1912.00757* (2019).

[72] Qiming Wang and Raul Castro Fernandez. 2023. Data Discovery using Natural Language Questions via a Self-Supervised Approach. *arXiv preprint arXiv:2301.03560* (2023).

[73] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[74] Gerhard Weikum. 2021. Knowledge graphs 2021: A data odyssey. *Proceedings of the VLDB Endowment* 14, 12 (2021), 3233–3238.

[75] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, Fabian Suchanek, et al. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations and Trends® in Databases* 10, 2-4 (2021), 108–490.

[76] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[77] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314* (2020).

[78] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2019. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311* (2019).

[79] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534* (2023).

[80] Zhen Zhang, Shuai Zheng, Yida Wang, Justin Chiu, George Karypis, Trishul Chilimbi, Mu Li, and Xin Jin. 2022. MiCS: near-linear scaling for training gigantic model on public cloud. *arXiv preprint arXiv:2205.00119* (2022).

[81] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, and Shen Li. 2023. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. arXiv:2304.11277 [cs.DC]

[82] Zixuan Zhao and Raul Castro Fernandez. 2022. Leva: Boosting Machine Learning Performance with Relational Embedding Data Augmentation. In *Proceedings of the 2022 International Conference on Management of Data*. 1504–1517.

[83] Shoshana Zuboff. 2015. Big other: surveillance capitalism and the prospects of an information civilization. *Journal of information technology* 30, 1 (2015), 75–89.