Building a Shared Conceptual Model of Complex, Heterogeneous Data Systems: A Demonstration

Michael R. Anderson[†*], Yuze Lou[†*], Jiayun Zou[†*], Michael Cafarella[‡], Sarah Chasins[§], Doug Downey[◇], Tian Gao[†], Kexin Huang[†], Dinghao Shen[†], Jenny Vo-Phamhi[†], Yitong Wang[†], Yuning Wang[†], Anna Zeng[‡]

[†]University of Michigan, [‡]MIT, [§]University of California, Berkeley, [◇]Allen Institute for Artificial Intelligence

ABSTRACT

The world of data objects and systems is complex and heterogeneous, making collaboration across tools, teams, and institutions difficult. Important goals like effective data science, responsible data governance, and well-informed data consumption all require participation from multiple parties who share conceptual data models despite being unfamiliar with, or organizationally distant from each other. In order to be productive together, data collaborators need a shared conceptual model that includes traditional schemas and system models, such as pipelines and procedures. This shared model does not have to be entirely correct, but to enable effective collaboration, it should be tool-, team-, and institution-independent. We describe a working demonstration system that aims to build this shared conceptual model. This system borrows ideas from knowledge graphs and other massive collaborative efforts to curate data artifacts beyond the reach of any one person or institution.

1 INTRODUCTION

The world of data systems is complex and heterogeneous, and getting more so. Organizations have moved far past the time of consolidating information in a single relational database; instead, data management work takes place across a dizzying array of databases, servers, laptops, data lakes, bulk processing systems, cloud storage, cloud-hosted applications, web services, user-facing apps, polystores, graph databases, and machine learning services.

This heterogeneity poses a huge problem for collaborative work that requires shared conceptual models of data and computation procedures.

The lack of shared models:

- Makes **data science** less productive, as scientists cannot easily rely on standard data definitions and operations.
- Makes **data governance** frustrating, as organizations cannot automatically enforce rules that apply to all of their employees' data activities.
- Makes **data consumption** tedious, as individuals can never know exactly the assumptions that went into a particular report or visualization.

We need a single shared model to abstract away the mountains of practical details making modern data systems possible at the systems level but nearly unmanageable at the semantic level. Perhaps such a model could incorporate not just traditional relational schemas, but also descriptions of shared datasets, functions, pipelines, and even provenance relationships between data objects, even across tools and institutions.

With such a model:

- **Data scientists** could rapidly converge on shared datasets, schemas, function implementations, data quality tests, and other primitives. They could quickly examine details of upstream inputs and downstream data consumers.
- **Data governance systems** could rely on the existence of correct provenance for any data object, regardless of where it is found. This could enable straightforward enforcement of General Data Protection Regulation (GDPR) usage restrictions, the GDPR right to be forgotten, the California Data Protection Act, and corporate sharing rules.
- **Data consumers** could investigate the unambiguous details of how any data output was generated, regardless of where the object or the user sits. This would allow for a decision-maker or news consumer to carefully and responsibly use aggregated results.

But how can we possibly agree upon and construct such a model?

Current solutions — Unfortunately, conventional solutions for creating shared semantic models have not been successful in today's heterogeneous environment. Traditional relational databases clearly only capture a small fraction of all data activity. XML-driven schema standards have failed to become popular and practical outside a relatively small number of uncontroversial and static domains that enjoy very wide consensus, such as addresses. Data catalog systems, such as Alation, Collibra, or data.world have become popular in recent years and are perhaps the most successful. However, users commonly report that: (1) data catalog systems only capture a fraction of data activity, (2) the manual curation workload required by these systems places an expensive limit on how quickly the catalog can grow, and (3) even high-quality catalogs do not generate large usage outside legally-mandated activities.

Moreover, even these catalog systems fail to capture lineage or provenance information. This is a growing area of research interest, but deployed systems are rare.

Collaborative data construction — Building such a comprehensive shared model may sound nearly impossible, but we have real-world examples of collaborative systems that have yielded high-quality and inexpensive data artifacts: PageRank-driven search engines like Google Search, social content curation systems like Reddit, Facebook, Pinterest, and Urban Dictionary, and — most notably for us — crowdsourced knowledge graphs like Wikidata.

These examples do not rely entirely or even primarily on traditional database ideas of good manual schema design or data integration quality. Rather, they combine three basic design elements:

[*]These authors contributed equally to this submission.

(1) **Broad collection** of raw information (such as web pages with hyperlinks, reactions and comments on online forums, or unexamined fact triples) by independent users.
(2) **Social ranking and aggregation** methods that exploit use phenomena to forge some form of consensus over these objects (such as a single PageRank score for every web page, or a ranking of popular news articles, or a set of deduplicated knowledge graph properties). Crucially, this software can often succeed even with imperfect semantic insight into the objects.
(3) **Presentation tools** that customize the socially-aggregated results and make them useful for individuals (such as term-weighted text search, or topic filters on a social media feed, or a voice agent that finds and renders user-requested facts). By channeling use toward highly-ranked items, these tools drive further consensus.

Put another way, existing social systems have succeeded at building datasets that are comprehensive but

Our demonstration — Building a comprehensive shared model of the data world will be a difficult and lengthy effort that involves large numbers of people; it is not even close to being done. However, we have built a demonstration system that aims to enable the construction of a very simple version of this shared model. It embodies the three above design elements.

In this paper we first discuss a few collaborative data systems that have been used to create similar artifacts. We then describe the demonstration system: its architecture and data model, a detailed user walkthrough, and ideas on how to make its deployment and sustained growth a practical effort. Finally, we conclude with some proposed research directions that would make this system more realistic and useful.

2 DESIGN INSPIRATION: COLLABORATIVE DATA CONSTRUCTION

In this section we briefly describe how Wikidata, a knowledge graph project, provides a compelling example of how online collaboration can be used to create high-quality data artifacts at a reasonable cost. We also briefly discuss two other systems: PageRank-driven search and social content curation.

2.1 Wikidata

Wikidata [19] is a knowledge graph (KG) that provides the structured data elements of most Wikipedia pages, often shown on the right-hand side of the page. Other knowledge graph examples include DBpedia [1], the Google Knowledge Graph [15]), UniProt [18], MusicBrainz [14], GeoNames [8], and many others [3, 7, 17]. Knowledge graphs have had slightly different definitions over the years. We think of a knowledge graph as a data resource which contains:

- *Unique entities* that correspond to real-world objects.[1] For example, entity Q76 represents Barack Obama in Wikidata. Different KGs make different curatorial decisions about what entities should be contained. For example, there is a Joe Biden entity in Wikidata, but not in the MusicBrainz graph of recorded music. [2] These can be thought of as the nodes in the knowledge graph.
- *Unique properties* that describe a directed relationship between two entities or an entity and a literal data value. For example, Wikidata property P19 describes the *place of birth* relationship, which describes an entity (usually a person) that is linked to another entity (usually a location). In contrast, Wikidata's property P569 (*date of birth*) usually describes a relationship between a single entity and a date. These properties can be thought of as potential edge labels in the knowledge graph.
- By combining entities, properties, and data values, the knowledge graph can hold a large number of facts about real-world objects. For example, Wikidata states that (Q76, P26, Q13133) is true. That is, Barack Obama (Q76) *has spouse* (P26) of Michelle Obama (Q13133). These *triples* can be thought of as concrete node-edge-node patterns in the knowledge graph.

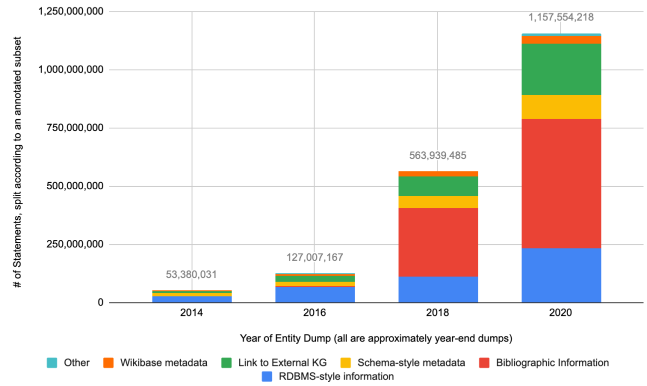

Figure 1: Growth in Wikidata size from 2014 to 2020

Although Wikidata is graph-structured, it is not substantially focused on capturing graph-oriented data, such as social networks or air routes. A significant portion of its data would be an easy fit for the relational model.

Growth and costs — Wikidata has enjoyed jaw-dropping growth. Figure 1 shows the growth in the number of fact triples in Wikidata, across multiple different fact types. The cost of building this quickly-growing dataset is not easy to quantify. One possible metric is the number of edits that have gone into the system during this time. Figure 2 shows the number of edits, stratified by users and user-deployed bots, that have created the growth in Figure 1. It is difficult to say whether this number is "cheap" or "expensive," but at least we can say that there is no super-linear growth in administration costs. Wikidata has effectively recruited increasing amounts of editor effort to produce increasing amounts of data.

Social curation — "Schema"-like collections of properties exist for some kinds of nodes (for example, Humans (entity Q5) generally

[1] Some academic knowledge networks follow a slightly different approach, such as creating a node for every distinct noun phrase in a text, even if they refer to identical real-world objects, as in VerbKB ([21]). However, our definition here is consistent with the major deployed knowledge graphs.

[2] Barack Obama appears in both, having recorded several audiobooks.

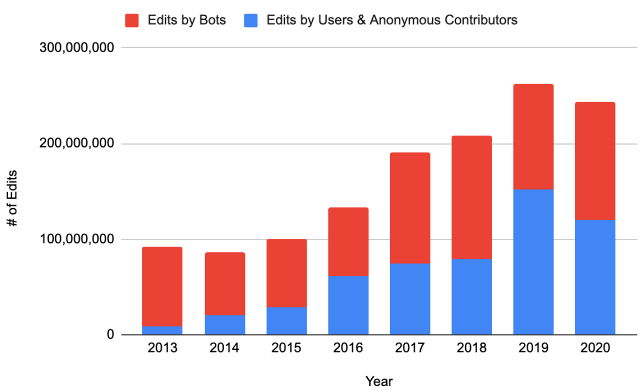

Figure 2: Wikidata edits from 2014 to 2020

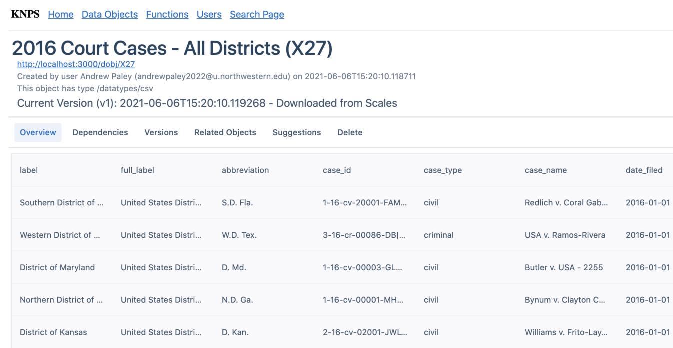

Figure 3: A KNPS database that describes federal court cases.

generally have a date of birth (P569), occupation (P106), and country of citizenship (P27)), but populating such facts for a given entity is not required or computationally enforced as with a standard relational database. Yet Wikidata is typically able to obtain good data quality. *Permissive admission* means that simple new facts do not receive much scrutiny before being added. Editor time is instead mainly spent evaluating facts that employ novel properties. *Aggressive autocomplete* software continually recommends the data entry user replace a partially-entered entity name with a popular object already in its dataset, and does the same for properties. Casual editors can thereby add data without mistakenly introducing unneeded and duplicative entities or properties.

2.2 Other Systems

We can now compare Wikidata's methods to other well-known examples of collaborative data construction.

PageRank, beyond being a signal for search engines, can be viewed as a collective effort to construct a global ranking of all known web pages. This ranking would be impossible for any one human to construct; it reflects a vast number of points of view, yet is generally viewed as a high-quality and useful artifact.

PageRank "contributors" add links to the system by publishing links on their own web pages that can be crawled and deduplicated. The PageRank algorithm then aggregates these link-votes into a consensus ranking. By using PageRank as an ingredient in a public search engine, users are further encouraged to view and share popular sites, driving further consensus in future PageRank outputs.

Social Content Curation systems like Reddit, Facebook, YouTube, and TikTok similarly aggregate user activity to construct high-quality rankings (of URL-addressable pieces of content). Contributors add a link to a shared repository. The social content system deduplicates the objects, then uses both explicit and implicit per-URL votes — revealed by users' reading, responding, and sharing behavior — to build a ranking over the objects. Again, by using the ranking to guide users' behavior toward popular objects, the system drives additional consensus on the "best" content objects.

2.3 Design Discussion

It is easy to wonder whether these systems are actually delivering high-quality artifacts. Wikidata seems to be very accurate at the human-inspectable fact level, but the best measure of quality would require evaluating a query workload. Unfortunately, the most popular Wikidata-powered workloads — voice agents and structured web search — are not easy to evaluate outside a few tech giants that have access to query logs. PageRank's consensus has arguably led to a small number of sites capturing almost all user attention. Social content systems' consensus rankings are popular but may be more inflammatory than actually high-quality.

We argue that in today's heterogeneous environments, a *broad-but-flawed consensus* picture of the world of data objects would yield dramatic steps forward for our collaborative semantic use cases, as it has for Wikidata, PageRank, and the other systems described. Data scientists could implicitly standardize their work around a relatively small number of shared datasets, thereby avoiding a huge amount of data prep work. Data governance administrators could reason that most company data fits a handful of broadly-accepted schemas, and thus write enforcement rules that are relevant to most company data. Data consumers could examine reports and find they were created with a small handful of widely-shared and debugged methods, and thereby not worry that their conclusions were driven by bugs in statistical code.

One deep challenge with this approach, for which we today only have a weak answer: traditional data models, such as relational schemas, generally need to be entirely correct, so how can an always-imperfect model ever hope to be useful? For now, we will aim to build a shared model that is good enough for humans to consult, but probably not good enough for most software to consume directly. An interesting direction for future research would be a novel type of shared data model that permits use by query tools even when the model is flawed.

3 DEMONSTRATION SYSTEM

We can now describe our concrete demonstration system. We first describe some system basics and its data model, then illustrate it with a user walkthrough, and finally describe some deployment practices to make the large consensus model a reality.

Figure 4: An analytical result derived from the judicial database, represented forever under a unique KNPS identifier.

3.1 System Basics

Just as the Wikidata knowledge graph models general-interest objects, and as MusicBrainz models the world of recorded music, the Knowledge Network Programming System (KNPS) aims to build a model of the data systems objects: files, databases, functions, schemas, images, pipelines, users, and so on. Edges in this graph represent relationships between objects: perhaps a User *created* a File, or a Database *ran-filter* to create a second Database. Fact triples can be added into the system by both social and automated means. For example, a user might explicitly upload a File; also, a filesystem crawler might automatically upload a File description. As with current KGs, the system does not impose sharp limits on what kinds of nodes or properties can be admitted; rather, it aims to build a fact set that is as correct and complete as possible.

KNPS differs from typical knowledge graphs in one critical way: users and automated processes can execute Function objects in the graph. Doing so will create new objects that are themselves stored in the graph. Current entity types in KNPS include CSVs, images, JSON files, PDFs, functions, and relational schemas. Like a traditional knowledge graph, adding new types is straightforward.

Note that KNPS's graph does not have the same intended semantics as a cloud database or a shared filesystem. Rather than being an always-reliable source of factual truth, it is expected that KNPS's graph will always be somewhat incomplete and incorrect. However, much like Wikidata's imperfect picture of the world, or a web crawl's imperfect picture of online content, KNPS aims to be close enough to correct to enable user progress (in this case on collaborative semantic projects). As mentioned above, we do not have a proposal for a data model that can be flawed and yet still directly usable by software. For now, our demonstration's notion of "user progress" always comprises direct user consumption of the shared graph we aim to build.

3.2 Walkthrough

We now present a short narrative of using KNPS.

Step 1. User Andrew from Northwestern has created an entry that describes a database about the US court system in 2016. The webpage that describes this entry is shown in Figure 3. The upper-left corner of the page shows metadata that is stored for any KNPS object: its unique identifier, the creator, creator's institution, creation date, title, and so on. In this case, the user has uploaded the database's entire contents, but doing so is not required (and in some cases may not be possible). There is no conceptual limit to the number of objects that can be created; if successful, the system should be able to handle on the order of hundreds of billions.

Sharing this database with a colleague is easy: the user simply forwards the URL. Like web pages under PageRank, we expect that some small number of KNPS objects will become popular and widely-used, but most will remain obscure.

The middle of Figure 3 shows the raw data content: the names of cases, whether they are criminal or civil, their duration, and so on.

Step 2. User Jiayun from Michigan has created a new entry, seen in Figure 4. This is an analytical result derived from the database in Figure 3. It shows the average duration of cases in federal districts in New York state. As above, it has a unique identifier that is intended to last forever. Creating this data object involved running a SQL query against object X27; because this query was run by KNPS, it was easy to automatically add the relevant provenance-style graph properties linking X27 and X36.

This aggregate query is interesting, but is a bit dry.

Step 3. User Mike from MIT has created the visualization — KNPS object X39 — seen in Figure 5. It is a chloropleth visualization of the result from object X36. This view of the object shows both the image and its provenance. This provenance graph was computed by following incoming provenance-related edges in the KNPS knowledge graph. Every node represents an object in the KNPS graph; the edges are a subset of the available graph properties.

Even this simple visualization required a range of inputs to build:

- At the upper-left, the "Case Duration for New York Courts by District" node is object X36.
- At the upper-right, "US Judicial Districts by County" is a dataset that maps from the names of judicial districts to county names. This was combined with the above object via the "Join CSV" stored function (itself a KNPS node). KNPS ran this function inside a hosted Singularity container.
- The "FIPS Codes for US Counties" node represents a dataset that maps from county labels to the numerical FIPS identification system. This was combined with the above intermediate result with the "Add FIPS" stored function (again, another KNPS node).
- Near the lower-right, "GeoJSON US Country FIPS data" maps from numerical FIPS identifiers to geographic polygons. When combined with the preceding data via the "Chloropleth Map" function, it yielded object X39.

Constructing this map required four datasets (the original judicial data, plus three on the way to the visualization) and involved at least three people from three different institutions. Of course, this could have been performed by standard tools available today, with files shared via email attachments. But since it was done via KNPS:

- A **data scientist** can examine the upstream provenance to see how the visualization was generated. The scientist can then reuse portions of this work — either the code or the auxiliary datasets — in the future.

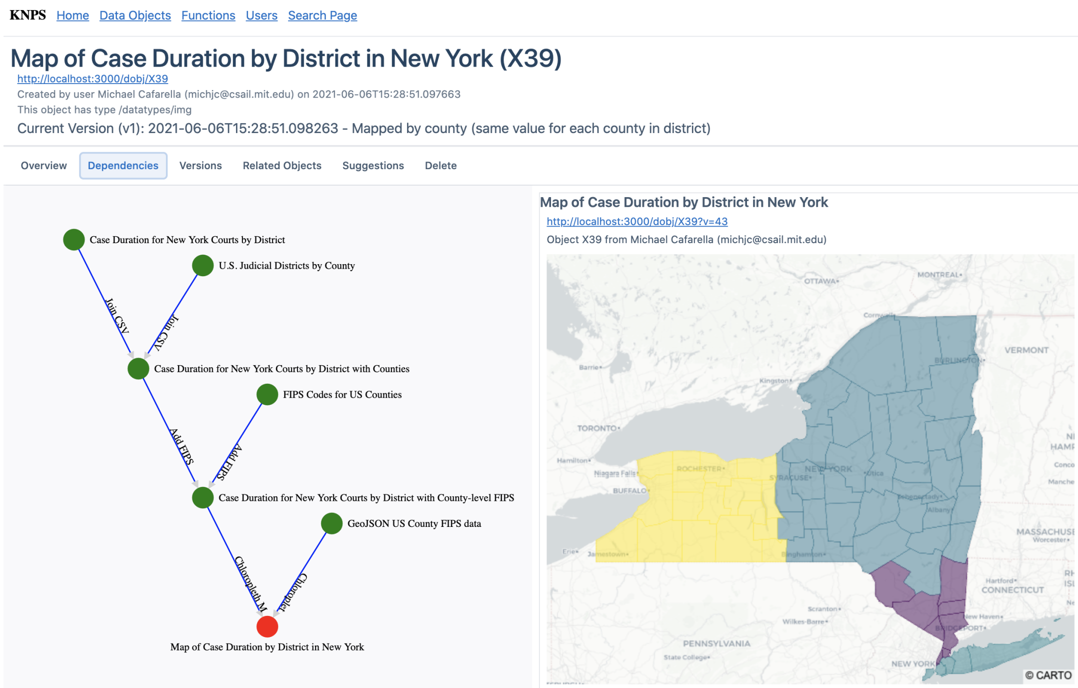

Figure 5: Visualization of the analytical result, with captured provenance

- A **governance system** can ensure that all of the visualization's inputs were datasets the organization is legally entitled to use.
- An informed **data consumer** can verify that the results reflect queries on high-quality datasets.

In contrast, in a traditional workflow, any metadata would have stopped at each institutional boundary. Even if the metadata had somehow been preserved and communicated to every user involved, a lack of a common vocabulary of data objects and functions would slow collaborative progress. The shared conceptual model is what makes effective collaboration possible.

We have also tested the system on a range of workflows, including an end-to-end implementation of the CORD-19 information extraction pipeline, which transforms raw scientific papers in PDF format into a bibliographic knowledge base of published coronavirus research [20].

3.3 Deployment Plans

Our demonstration system shows the value of using the KNPS graph. However, our narrative above shows a set of collaborating users who intentionally upload data and code to the system. For users willing (and able) to do so, a KNPS-contained collaboration environment will be useful. However, we realize that for various reasons, many users cannot be expected to perform the uploads needed to take advantage of KNPS's strict provenance features. Possibly, only the most motivated users will explicitly tell KNPS about their data objects. This will be disappointing: the system's value lies in its universality.

As a result, KNPS also allows for automated data upload and curation, emulating social content systems by allowing automatic broad data collection. For example, client software can automatically scan laptops, databases, or Amazon S3 buckets. A single node in KNPS can be potentially discovered by observing changes on a concrete local filesystem. A sharing event between two users can be potentially discovered by observing one user's bytes appear identically in another user's Downloads directory. Provenance events can be potentially recovered by watching local process lists or logs.

This model of collection is far messier than explicit user uploads, but likelier to obtain high recall. It will yield a large number of low-quality objects and may yield spurious edges between them. Like Google's crawl-and-PageRank system, KNPS will be permissive during the "crawling" data collection period, then engage in a substantial amount of post-collection cleanup, such as object deduplication. Finally, the graph can be used in the same ranking-consensus process described above.

We are building this collection system now. While we can demonstrate it, we do not yet know how effectively it will gather KNPS data. Because it lacks details that would be available with explicit user attention — for example, the exact semantics of functions — the resulting system may need to be more "semantically humble" than the walkthrough above suggests. This represents what we think is a core design tradeoff between expressiveness and concrete observed data quality.

4 RESEARCH DIRECTIONS

There are at least two lines of research that would enormously improve the value of KNPS. First, as mentioned in the section immediately above, automatic provenance capture would make the KNPS dataset larger and more useful. The core idea here is to use zero-labor commodity instrumentation methods, like watching process lists and web service logs, to infer higher-abstraction operations that are easier for humans to understand. Ideally, this "named operation recognition" task succeeds even when the operation comprises multiple independent binaries across machines.

A second longer-term project would be to investigate a shared data model that serve as the basis for query processing and optimization, while still containing the errors that we believe are an inevitable part of the social curation approach. For example, queries written against the KNPS consensus model should be able to run to some form of imperfect completion when a target dataset does not perfectly match the consensus. This direction is best explored after we understand the distribution of KNPS errors in practice.

5 RELATED WORK

Our system has some similarities to recent curation systems built to address problems in industrial machine learning deployment [11][4]. These go beyond standard packages of ML training algorithms to include data management, data transformation, versioning, and other features that make the end-to-end data experience easier. Unlike those systems, but like the Dataverse project [5], our system is intended for general-purpose and cross-institutional use. Unlike all of the above systems, we aim to emulate the design of the social collaboration systems described in Section 2

There has been a substantial amount of research in data provenance — or, relatedly, data lineage — in a database or reproducibility setting ([2, 6, 9, 10, 12, 13, 16]). Unfortunately, there is not yet a widely-adopted system in which provenance plays a major role. Explanations for why these systems are not widely adopted are potentially instructive. Existing systems require either substantial amounts of human effort to use or require adoption of a new tool; in either approach, a large amount of user activity goes uncaptured. In contrast, search and other social curation systems capture as much imperfect information as possible, then fix it up after the fact with a combination of ML- and socially-driven measures. Our demonstration system follows this second design approach.

6 CONCLUSIONS

We have described a system for building a shared conceptual model of heterogeneous and complex data systems. We believe this shared model is useful for a range of collaborative semantic applications: effective data science, responsible data governance, and informed data consumption. We used existing collaborative systems, especially Wikidata, as design inspiration for our demonstration system.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference* (Busan, Korea) *(ISWC'07/ASWC'07)*. Springer-Verlag, Berlin, Heidelberg, 722–735. http://dl.acm.org/citation.cfm?id=1785162.1785216

[2] Louis Bavoil, Steven Callahan, Carlos Scheidegger, Patricia Crossno, Claudio Silva, and Juliana Freire. 2005. VisTrails: Enabling Interactive Multiple-View Visualizations. *IEEE Visualization*, 18. https://doi.org/10.1109/VISUAL.2005.1532788

[3] Christian Bizer. 2009. The Emerging Web of Linked Data. *IEEE Intelligent Systems* 24, 5 (Sept. 2009), 87–92. https://doi.org/10.1109/MIS.2009.102

[4] Eli Brumbaugh, Mani Bhushan, Andrew Cheong, Michelle Gu-Qian Du, Jeff Feng, Nick Handel, Andrew Hoh, Jack Hone, Brad Hunter, Atul Kale, Alfredo Luque, Bahador Nooraei, John Park, Krishna Puttaswamy, Kyle Schiller, Evgeny Shapiro, Conglei Shi, Aaron Siegel, Nikhil Simha, Marie Sbrocca, Shi-Jing Yao, Patrick Yoon, Varant Zanoyan, Xiao-Han T. Zeng, and Qiang Zhu. 2019. Bighead: A Framework-Agnostic, End-to-End Machine Learning Platform. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 551–560. https://doi.org/10.1109/DSAA.2019.00070

[5] Mercè Crosas. 2011. The Dataverse Network®: An Open-Source Application for Sharing, Discovering and Preserving Data. *D Lib Mag.* 17, 1/2 (2011).

[6] Chenyun Dai, Dan Lin, Elisa Bertino, and Murat Kantarcioglu. 2008. An Approach to Evaluate Data Trustworthiness Based on Data Provenance. In *Secure Data Management*, Willem Jonker and Milan Petković (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 82–98.

[7] Oren Etzioni, Michael J. Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*. 100–110. https://doi.org/10.1145/988672.988687

[8] GeoNames 2019. GeoNames. http://www.geonames.org/. http://www.geonames.org/ [Online; accessed May 30, 2019].

[9] Todd J. Green, Grigoris Karvounarakis, Nicholas E. Taylor, Olivier Biton, Zachary G. Ives, and Val Tannen. 2007. ORCHESTRA: Facilitating Collaborative Data Sharing. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data* (Beijing, China) *(SIGMOD '07)*. Association for Computing Machinery, New York, NY, USA, 1131–1133.

[10] Olaf Hartig and Jun Zhao. 2009. Using Web Data Provenance for Quality Assessment. In *Proceedings of the First International Conference on Semantic Web in Provenance Management - Volume 526* (Washington DC) *(SWPM'09)*. CEUR-WS.org, Aachen, DEU, 29–34.

[11] Jeremy Hermann and Mike Del Balso. 2017. Meet Michelangelo: Uber's Machine Learning Platform. Accessed June 9, 2021. https://eng.uber.com/michelangelo-machine-learning-platform/.

[12] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. A survey on provenance: What for? What form? What from? *The VLDB Journal* 26 (10 2017). https://doi.org/10.1007/s00778-017-0486-1

[13] Sanjay Krishnan, Jiannan Wang, Michael J. Franklin, Ken Goldberg, and Tim Kraska. 2016. PrivateClean: Data Cleaning and Differential Privacy. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 937–951. https://doi.org/10.1145/2882903.2915248

[14] MusicBrainz 2019. Welcome to MusicBrainz! https://musicbrainz.org/. [Online; accessed May 30, 2019].

[15] Amit Singhal. 2012. Introducing the Knowledge Graph: things, not strings. https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html. https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html [Online; accessed May 30, 2019].

[16] Holger Stitz, S. Luger, Marc Streit, and N. Gehlenborg. 2016. AVOCADO: Visualization of Workflow-Derived Data Provenance for Reproducible Biomedical Research. *Computer Graphics Forum* 35 (06 2016), 481–490. https://doi.org/10.1111/cgf.12924

[17] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. ACM, New York, NY, USA, 697–706. https://doi.org/10.1145/1242572.1242667

[18] TheUniProtConsortium. 2018. UniProt: a worldwide hub of protein knowledge. In *Nucleic Acids Research*.

[19] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57, 10 (Sept. 2014), 78–85. https://doi.org/10.1145/2629489

[20] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Association for Computational Linguistics, Online. https://www.aclweb.org/anthology/2020.nlpcovid19-acl.1

[21] Derry Tanti Wijaya and Tom M. Mitchell. 2016. Mapping Verbs in Different Languages to Knowledge Base Relations using Web Text as Interlingua. In *NAACL HLT 2016, Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. 818–827. http://aclweb.org/anthology/N/N16/N16-1096.pdf