

Infrastructure for Rapid Open Knowledge Network Development

Running Title: Knowledge Network Infrastructure

Coauthor Details:

Michael Cafarella, MIT CSAIL, michjc@csail.mit.edu
Michael Anderson, University of Michigan, mrande@umich.edu
Iz Beltagy, Allen Institute for Artificial Intelligence, beltagy@allenai.org
Arie Cattan, Allen Institute for Artificial Intelligence, arie.cattan@gmail.com
Sarah Chasins, University of California, Berkeley, schasins@cs.berkeley.edu
Ido Dagan, Allen Institute for Artificial Intelligence, dagan@cs.biu.ac.il
Doug Downey, Allen Institute for Artificial Intelligence, dougd@allenai.org
Oren Etzioni, Allen Institute for Artificial Intelligence, orene@allenai.org
Sergey Feldman, Allen Institute for Artificial Intelligence, sergey@allenai.org
Tian Gao, University of Michigan, gaotian@umich.edu
Tom Hope, Allen Institute for Artificial Intelligence, tomh@allenai.org
Kexin Huang, University of Michigan, huangkex@umich.edu
Sophie Johnson, Allen Institute for Artificial Intelligence, sophiej@allenai.org
Daniel King, Allen Institute for Artificial Intelligence, daniel@allenai.org
Kyle Lo, Allen Institute for Artificial Intelligence, kylel@allenai.org
Yuze Lou, University of Michigan, yuzelou@umich.edu
Matthew Shapiro, University of Michigan, shapiro@umich.edu
Dinghao Shen, University of Michigan, jackyy@umich.edu
Shivashankar Subramanian, Allen Institute for Artificial Intelligence, shivashankar@student.unimelb.edu.au
Lucy Lu Wang, Allen Institute for Artificial Intelligence, lucyw@allenai.org
Yuning Wang, University of Michigan, wyuning@umich.edu
Yitong Wang, University of Michigan, yitongw@umich.edu
Daniel S. Weld, Allen Institute for Artificial Intelligence, danw@allenai.org
Jenny Vo-Phamhi, University of Michigan, jennyvo@umich.edu
Anna Zeng, MIT CSAIL, annazeng@mit.edu
Jiayun Zou, University of Michigan, alicezou@umich.edu

Acknowledgments: We would like to acknowledge the National Science Foundation Convergence Accelerator program for supporting this work.

Statements: This submission meets the author responsibilities listed in the AAAI "Publication Ethics and Malpractice Statement" of August 2017. This work was supported primarily by a grant from the National Science Foundation.

ORCID: 0000-0001-6122-0590

Author Manuscript

Infrastructure for Rapid Open Knowledge Network Development

an Gao, Tom Hope, Kexin Huang, Sophie Johnson, Daniel King, Kyle Lo, Yuze Lou, Matthew Shapiro, Dinghao

Author Manuscript

The past decade has witnessed a growth in the use of knowledge graph technologies for advanced data search, data integration, and query-answering applications. The leading example of a public, general-purpose open knowledge network (*aka* knowledge graph) is Wikidata, which has demonstrated remarkable advances in quality and coverage over this time. Proprietary knowledge graphs drive some of the leading applications of the day including, for example, Google Search, Alexa, Siri, and Cortana. Open Knowledge Networks are exciting: they promise the power of structured database-like queries with the potential for the wide coverage that is today only provided by the Web. With the current state of the art, building, using, and scaling large knowledge networks can still be frustratingly slow. This article describes a National Science Foundation Convergence Accelerator project to build a set of Knowledge Network Programming Infrastructure systems to address this issue.

Keywords: knowledge network, knowledge graph, infrastructure, extraction, synthesis

The growth of the Wikidata open knowledge network is one of the remarkable stories of the past decade of computing. The Wikidata OKN (Vrandečić and Krötzsch 2014), which supplies structured components of Wikipedia and is also used to power voice agents and structured search applications, grew from roughly 53M factual statements in 2014 to more than 1.1B in 2020 (Zeng, Sabek, and Cafarella 2021). Other knowledge graph examples include DBpedia (Auer et al. 2007), the Google Knowledge Graph (Singhal 2012), UniProt (TheUniProtConsortium 2018), MusicBrainz (MusicBrainz 2019), GeoNames (GeoNames 2019), and many others (Suchanek, Kasneci, and Weikum 2007)(Etzioni et al. 2004) (Bizer 2009).

Unfortunately, with the current state of technology, complex and large knowledge network structures can be tedious to construct, refine, and use.

Our Knowledge Network Infrastructure NSF Convergence Accelerator project aims to address this situation by providing data-oriented software infrastructure for efficient and effective management of OKNs during their *construction*, *on-going refinement*, and *use*, as described below. The project is a collaboration among researchers at MIT, the University of Michigan, the Allen Institute for Artificial Intelligence, the University of California at Berkeley, and the University of Washington.

OKN Construction --- An important source of OKN data is human-readable documents. We have constructed an OKN called CORD-19, which comprises information on the coronavirus, and was first released in March of 2020, shortly after the virus became widely known. While extracting information from text has been a long-standing research area, we demonstrate new techniques that make two traditional information extraction approaches, *viz.*, document processing and author disambiguation, dramatically faster.

OKN Refinement: While large sections of OKNs could be derived from large-scale ML processes, it is also important to facilitate narrow, immediate, use case-driven refinements from users with little extraction-related training. We discuss a novel program-synthesis method that allows non-programmers to rapidly build novel data ingestors.

OKN Use: OKN use is perhaps the broadest of these challenges. Many use cases (in using them for data science processes, or data governance rules, or just informed data consumption) are slow and tedious because the OKN creation procedure is opaque. We propose a new software architecture for building OKNs that simultaneously builds an annotated and shareable model of the construction procedure, enabling users to more quickly put OKNs to good use.

The following three sections describe project activities and advancements in the above three areas: OKN Construction, OKN Refinement, and OKN Use.

OKN Construction: CORD-19

Our project has developed functionality to assist in the rapid construction of OKNs. We will describe the work in the context of the well-known CORD-19 dataset. In March of 2020, the Allen Institute for AI, in collaboration with The White House Office of Science and Technology Policy and others¹, released the CORD-19 dataset, a knowledge graph of publications and preprints on COVID-19 and related topics (Wang et al. 2020). Notably, its initial release was just a few months after the COVID-19 virus became known. CORD-19 and its infrastructure components, known as Semantic Scholar, have yielded a range of lessons for our project on how to improve the speed at which such knowledge graph structures can be constructed. We first provide an overview of the CORD-19 dataset, followed by brief descriptions of two functions/tools that we have developed to speed the construction of OKNs: *layout-aware document processing* and *“low-labor” author disambiguation*.

Overview of the CORD-19 Dataset

The CORD-19 dataset includes papers from over 3,200 journals from both free, online sources and commercial publishers. It contains bibliographic data such as titles, authors, venues, and citations. It also includes full text content for more than a third of its papers. The dataset started with 28,000 papers in its first release, and has steadily grown to contain over 750,000. The resource is now updated weekly, moving from daily updates during its peak. The dataset has proved to be successful, with over 3.5 million views and the most upvotes of any data set in the history of the Kaggle platform, where it was hosted. CORD-19 has also been used to power the popular TREC-COVID shared task (Roberts et al. 2020) and a variety of new public visualization and search tools. A final exciting outcome of CORD-19 has been the wide variety of different derived products of the dataset created by independent groups from across the Web.² These derived data items include links to other knowledge graphs, named entity tags, and so on.

CORD-19’s rapid development allowed us to identify and address several concrete artificial intelligence tasks that are key but also traditionally very slow, as described below.

SciCo: Concept Coreference and Hierarchy

Consider a computer science researcher hoping to answer the question “which authors have written the most papers about pre-trained language models and text classification?” Beyond a bibliographic OKN of authors and papers as in CORD-19, to answer this kind of question a system would need knowledge drawn from the scientific content of the papers—saying which ones contain mentions of a “pre-trained language model” and a “text classification” task. Automatically acquiring this knowledge is challenging for multiple reasons. First, the system must determine which scientific entities (e.g. “RoBERTa”) fall into which categories (“pre-trained language model”), and no comprehensive ontology exists for these evolving concepts. Further, disambiguation is required, since different papers may refer to the same concept using different names, and likewise the same name may be used to refer to distinct concepts.

To begin to address this challenge we created SciCo, a data set for identifying cross-document coreference between methods and tasks in the computer science domain (e.g., that “named entity typing” and “named entity classification” are the same task), along with hierarchical relationships between those concepts (e.g., that a mention of “document classification” entails “text classification”) (Cattan et al. 2021). SciCo covers an instance of the recently popular task of *cross-document coreference*, except instead of focusing on coreference of events featuring concrete entities (e.g., people, locations) from the news domain (Cybulska and Vossen 2014), SciCo tackles abstract scientific concepts and hierarchical entailment relationships in addition to coreference. These characteristics allow SciCo to form a step toward answering the complex example query in the previous paragraph. Building on

(Cattan et al. 2020), we constructed a new annotation interface for our task, developed candidate selectors to bootstrap mentions for annotators to label, and trained expert annotators on the task. (The annotation interface will eventually be integrated into KNPS.) The resulting data set is three times larger than the comparable ECB+ data set for coreference on the news domain (Cybulska and Vossen 2014). Baseline algorithms score well below the level of annotator agreement on the hierarchical task, leaving substantial room for improvement, but we find that methods that consider both the hierarchical task and the coreference task jointly outperform disjoint baselines. In future work we hope to develop new joint approaches and also measure whether SciCo can improve question answering or faceted search in practice.

Layout-Aware Document Processing

Documents in important technical domains like science and law (as described in the paper by *Amaral et al* on the SCALES project in this special issue) often come in PDF format. We observed that information extraction from these documents can benefit from using the layout of the text, which often signal the semantics of terms. For example, key fields like a paper’s title and authors are offset from the main text, documents use tabular formats to signal relational data, and so on. Despite the importance of this signal, most existing NLP processing pipelines and tools have only considered raw text without considering document layouts.

However, recovering and exploiting document layout information for the CORD-19 dataset turned out to be a stumbling block to rapid OKN construction. First, the intensely visual qualities of layout information made collecting human annotations for training data a slow and burdensome process. Second, there were few existing transfer learning resources, akin to the various precomputed embeddings available for raw text.

We have addressed these layout-centric pinch points in the OKN construction process in two ways. First, we created PAWLS (PDF Annotation with Labels and Structure), a new annotation tool designed for PDF documents (Neumann, Shen, and Skjonsberg 2021). PAWLS supports labeling span-based textual regions, free form visual bounding boxes, and easy authoring of n-ary relations among different visual elements (see Figure 1). We are currently using PAWLS to label a large, new challenge set for extraction of bibliographic knowledge from scientific documents.

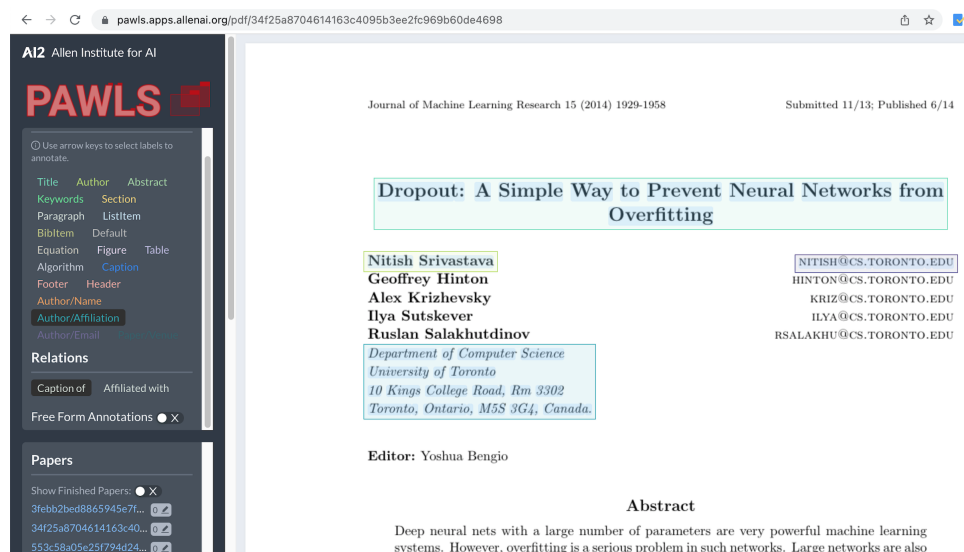


Figure 1: The PAWLS Interface for Labeling Regions of PDF Documents.

Second, we created *LayoutParser*, an open-source library for applying and customizing deep learning models for layout-aware tasks such as layout detection

and character recognition (Shen et al. 2021). *LayoutParser* also includes a platform for sharing pre-trained models and document digitization pipelines. We also developed an associated set of techniques for cost-effectively tailoring existing pretrained models like BERT or RoBERTa to scientific document layout without the need for the expensive additional layout-aware pretraining required by recent models. Our techniques rely on the simple idea that scientific layout typically involves visually-distinct groups of tokens (lines or blocks) that share the same semantic category (title, author, etc). One simple technique simply encodes these groups using indicator tokens in the model’s textual input, and we show how this technique is able to match the performance of the recent LayoutLM model (Xu et al. 2020) but with more than an order of magnitude lower training cost (Shen et al. 2021).

Low-Labor Author Disambiguation

Author disambiguation has been a long-standing problem in text understanding systems where the system is given a set of author mentions, including author names and the papers they are attached to, and the task is to cluster these into sets of mentions that represent the same real-world person.

This is a critical, difficult task faced by every bibliographic database. Even state-of-the-art systems may not achieve high quality results (Zhang et al. 2018). A number of algorithms have been proposed (Ferreira, Gonçalves, and Laender 2012). Unfortunately, comparing these algorithms is difficult because they have tended to be evaluated on disparate datasets using different features. As a result, simply choosing an appropriate author disambiguation strategy for a novel OKN is itself a time-consuming and exhausting task.

We addressed this challenge by introducing S2AND (Subramanian et al. 2021), a unified benchmark dataset for author disambiguation. S2AND coheres eight different datasets from the literature into a unified resource and a single feature set. Our analysis reveals that the previous data sets tend to cover idiosyncratic portions of the literature, which means that training on the combination of data in S2AND tends to provide stronger disambiguation performance when tested on held-out datasets. As a result, a reference implementation that we provide, when trained on S2AND, is able to achieve a 50% error reduction over the production author disambiguation algorithm used within Semantic Scholar. This has accelerated OKN construction in two ways: by reducing the work needed to evaluate a given algorithm, and by simply improving the baseline method, thereby reducing the salience of a previously-difficult design decision.

While S2AND’s features like author affiliations, co-authors, and paper embeddings (Cohan et al. 2020) can power reasonable performance using our reference implementation, there is still ample room for improvement, and we hope that the availability of S2AND spurs the development of new methods.

OKN Refinement

In our view, refining an OKN is a distinct task in the OKN lifecycle. For a knowledge network to be open and capable of supporting a range of concrete user-facing applications, it must be possible to refine the OKN to suit the purpose. It should not be necessary to run an entirely new data extraction pipeline from scratch, or generate a mass of new training data for relatively small refinements. It may also be necessary to fix inconsistencies, or other “bugs”, in the structure as part of the refinement stage. Our project aims to provide the necessary toolchains to efficiently support these tasks.

In particular, we focus on the problem of incremental data refinement of an OKN in the context of specific application use cases. An *ingestor*/incremental curation/refinement program may be burdensome for even a data expert to write, but possibly entirely beyond the reach of most non-technical people who may nonetheless be domain experts or expert app designers. One of our innovations is a lightweight web data ingestion synthesizer that lets non-programmers use data they find in web pages to quickly and incrementally refine an OKN.

The rapid OKN refinement system is designed around three design principles. First, it should allow users to quickly augment the OKN with web-derived datasets. We envision users turning to lightweight ingestion in the midst of OKN-focused work. For example, when developing a financial application, a user realizes 2020's exchange rate data for a particular currency pair has not been added to the OKN; they find the data on, say FRED, an online economic data repository, and import it into the OKN.

Second, it should be accessible to non-programmers. The system is *lightweight* in the sense that it provides a quick and low-effort method to add new datasets, requiring minimal training of ML models.

Third, the refinement programs should be customizable for future use to make this new extended OKN sustainable over the long haul.

In our approach, we treat the lightweight data refinement problem as two core challenges: (1) web data extraction and (2) OKN-to-dataset integration. On the web data extraction side, we take advantage of our team's prior work in programming-by-demonstration (PBD) tools (Chasins, Mueller, and Bodík 2018) (Barman et al. 2016). Users write web data collection scripts by demonstrating how to collect a sample of the data. For example, to demonstrate how to collect a year of exchange rate information, the user interacts with the webpage to highlight the first day of exchange rate information. On the OKN-to-dataset integration side, we take advantage of the incomplete data already available in the OKN. For example, if the tool notices that 70% of entities in a role are of a certain type, say, "mountains," then the tool can suggest filtering matches to include only mountain-labeled entities, or adding the mountain property to entities that lack it.

Bringing disparate datasets into an integrated knowledge network supports novel dataset integration tools for *ad hoc* and long term data integration. For *ad hoc* integration, access to many plausibly linkable datasets allows a synthesizer to suggest integration programs based on overlaps in the data and other tests of semantic similarity. In the longer term, building up a repository of integration programs and the data they link lets us build a training set for improving future variants of the synthesizer and directly reuse difficult but important integration snippets. The end result is to make it possible for non-coders to integrate the data they need from across many disparate sources without the tedious manual effort required today. This longer-term approach has a synergistic relationship with the software architecture we describe in the section below.

Facilitating OKN Use

As mentioned earlier, and in other articles in this special issue, large-scale knowledge networks like Wikidata are in use in some large consumer-style workloads like voice assistants and structured web search. However, other OKNs, such as CORD-19 and other domain-specific datasets discussed in this issue, suggest more technical data science-style use cases.

These data science activities need to be accompanied by high-quality data governance and better-informed data consumption by applications and users.

The Need for a Shared Data Processing Model

A core value proposition of OKNs is the ability to integrate data contributions from a range of sources. However, the heterogeneous and cross-institutional nature of this undertaking also raises hard challenges for downstream use cases. For example, in the case of CORD-19, the data scientists outside the CORD-19 team who create derived data products may not know when an upstream paper extraction is modified. Concerning governance, the CORD-19 publishers cannot enforce rules about the dataset's use, such as requiring downstream users to place derived data products in the public domain. Finally, a data consumer who sees a potential problem in a visualization derived from CORD-19 will not have an easy time determining if there was a mapping bug, an extraction error, or simply a problem in the original published academic paper.

In all of these cases, users require a shared model of the data production process itself. Data scientists need to have a cross-institutional provenance chain that links upstream paper modifications to their downstream data science results. Governance systems need to connect the policies of data publishers with the actions of data users who might be far away in both time and organizationally. Informed data consumption entails answering meaningful questions about how the data was created.

Today, creating this shared model is incredibly time-consuming and difficult. A version of it exists implicitly in close-knit teams with substantial amounts of shared knowledge. Some systems, like relational databases, effectively maintain their own understandable data models, but they are tool-specific. Existing data catalog systems do allow for systematic deposit of metadata, but they rarely stretch across institutions, and the cost of explicit manual curation is quite high.

Across tools, teams, and organizations, there is today no collaborative software system that allows data works to affordably create this shared model. Instead, they have no choice but to create it via "data archaeology": emails, phone calls, shared design documents, and other manual one-off efforts. The lack of an explicit cross-institutional model of data processes in most cases makes the usage side of OKNs perhaps the slowest of all of our three thrusts.

Any shared model of data production processes will need to balance fidelity against the need to be comprehensible to everyone. Like OKNs or any database, this model will be imperfect. However, we argue that a *broad-but-flawed consensus* picture of the world of data processes would dramatically accelerate the tasks we've described.

Somewhat ironically, the socially-driven Wikidata-style OKN production process is probably the best (although not the only) method we have for this kind of cross-institutional data creation. We describe a system we call the Knowledge Network Programming System, or KNPS, that attempts to use OKN-style methods to construct a shared picture of the data production process, and thereby accelerate all of the above *use-* oriented OKN tasks.

KNPS Basics

Just as the Wikidata OKN models general-interest objects, and as MusicBrainz models the world of recorded music, KNPS aims to build a model of data production processes: 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 OKNs, 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 OKNs 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 include CSVs, images, JSON files, PDFs, functions, and relational schemas. Like a traditional OKN, adding new types is straightforward.

Walkthrough

We can now present a short narrative describing how KNPS works today:

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 on the left of Figure 2. 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. The middle of the page shows the raw data content: the names of cases, whether they are criminal or civil, their duration, and so on.

2016 Court Cases - All Districts (X27)
 Created by user Andrew Hiley (andhiley@mit.edu) on 2021-06-06T15:20:10.18371
 The object has type `dataset/table`
 Current Version (v1): 2021-06-06T15:20:10.18371 - Downloaded from Scales

label	full_label	abbreviation	case_id	case_type	case_name	date filed
Southern District of ...	United States Dist...	S.D. Fla.	1:16-cv-02001-FAM...	civil	Reidlich v. Coral Dub...	2016-01-01
Western District of ...	United States Dist...	W.D. Tex.	3:16-cr-00080-DB...	criminal	USA v. Ramon Rivera	2016-01-01
District of Maryland	United States Dist...	D. Md.	1:16-cv-00003-GL...	civil	Butler v. USA - 2355	2016-01-01
Northern District of ...	United States Dist...	N.D. Ga.	1:16-cv-00001-MH...	civil	Bynum v. Clayton C...	2016-01-01
District of Kansas	United States Dist...	D. Kan.	2:16-cv-02001-JRL...	civil	Williams v. Frito-Lay...	2016-01-01

Case Duration for New York Courts by District (X36)
 Created by user Jiajun Zhu (jiajunzhu@mit.edu) on 2021-06-06T15:24:02.68487
 This object has type `dataset/table`
 Current Version (v1): 2021-06-06T15:24:02.685367 - Mean of Case Durations for each NY court district

abbreviation	case_duration
S.D.N.Y.	196.362990...
E.D.N.Y.	247.1327895...
N.D.N.Y.	241.689056...
W.D.N.Y.	311.7678923...

Figure 2: At left, a KNPS database that describes federal court cases. At right, an analytical result derived from the judicial database, represented forever under a different unique KNPS identifier

Step 2. User Jiayun from Michigan has created a new entry, seen in the right-hand image in Figure 2. This is an analytical result derived from the database on the left. 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 3. It is a choropleth 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 “Choropleth 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’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 the future, the system could even color-code upstream inputs according to how widely-used they are, in an effort to approximate reputation and trustworthiness.

In contrast, in a traditional workflow, any metadata would have stopped at each institutional boundary. This would have forced all of users back into the standard,

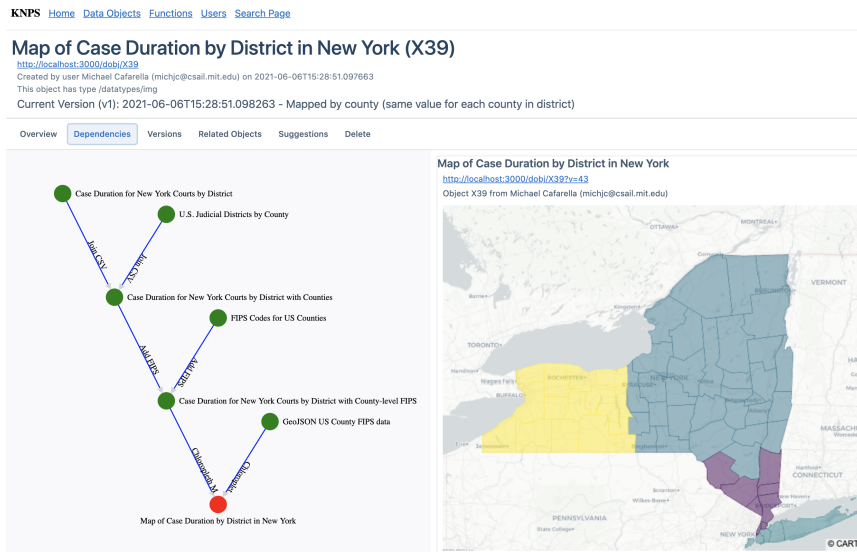


Figure 3: Visualization of the analytical result, with captured provenance

slow data archaeology process so common today. The shared data process model is what makes effective cross-institutional work fast.

We have tested the system on a range of workflows, including an end-to-end implementation of the COVID-19 pipeline.

Practical Deployment and “System Extraction”

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, KNPS will be a powerful tool for creating the shared model we think is needed for the full range of use applications. However, we realize that due to a number of reasons (data privacy, transfer challenges, and so on), many users cannot be expected to perform the uploads needed to take advantage of KNPS’s strict provenance features. This will be disappointing: the system’s value lies in its universality.

As a result, KNPS also allows for automated data upload and curation. 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 automatic approach will likely yield a more complete, but lower-quality, version of the shared data processing model. It also promises a strange and novel kind of information extraction research: deriving a complete, correct model of the planet’s history of data operations while observing only the partial and imperfect signals available via standard system instrumentation. We think this is an exciting new area for extraction research with a clear path to making OKN activity better.

Conclusions

Open knowledge networks have the potential to power many interesting and impactful applications by integrating data and information about real-world entities from a large and diverse variety of source. Harnessing the power of such a structure require infrastructure for OKN Construction, Refinement, and Use.

Acknowledgments

The authors would like to acknowledge the National Science Foundation and in particular the staff of the Convergence Accelerator program for their crucial support.

Notes

¹The National Library of Medicine, the Chan Zuckerberg Initiative, Microsoft Research, Kaggle, and Georgetown University's Center for Security and Emerging Technology

²<https://github.com/w3c/hcls/wiki/CORD-19-Semantic-Annotation-Projects>

References

- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 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, ISWC'07/ASWC'07*, 722--735. Berlin, Heidelberg: Springer-Verlag.
- Barman, S.; Chasins, S.; Bodík, R.; and Gulwani, S. 2016. Ringer: Web Automation by Demonstration. In Visser, E., and Smaragdakis, Y., eds., *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2016, part of SPLASH 2016, Amsterdam, The Netherlands, October 30 - November 4, 2016*, 748--764. ACM.
- Bizer, C. 2009. The Emerging Web of Linked Data. *IEEE Intelligent Systems* 24(5):87--92.
- Cattan, A.; Eirew, A.; Stanovsky, G.; Joshi, M.; and Dagan, I. 2020. Streamlining Cross-Document Coreference Resolution: Evaluation and Modeling. *arXiv preprint arXiv:2009.11032*.
- Cattan, A.; Johnson, S.; Weld, D.; Dagan, I.; Beltagy, I.; Downey, D.; and Hope, T. 2021. SciCo: Hierarchical Cross-Document Coreference for Scientific Concepts.
- Chasins, S. E.; Mueller, M.; and Bodík, R. 2018. Rousillon: Scraping Distributed Hierarchical Web Data. In Baudisch, P.; Schmidt, A.; and Wilson, A., eds., *The 31st Annual ACM Symposium on User Interface Software and Technology, UIST 2018, Berlin, Germany, October 14-17, 2018*, 963--975. ACM.
- Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2270--2282. Online: Association for Computational Linguistics.
- Cybulska, A., and Vossen, P. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *LREC*, 4545--4552.
- Etzioni, O.; Cafarella, M. J.; Downey, D.; Kok, S.; Popescu, A.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 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.
- Ferreira, A. A.; Gonçalves, M. A.; and Laender, A. H. 2012. A Brief Survey of Automatic Methods for Author Name Disambiguation. *SIGMOD Rec.* 41(2):15--26.
2019. GeoNames. <http://www.geonames.org/>. [Online; accessed May 30, 2019].
2019. Welcome to MusicBrainz! <https://musicbrainz.org/>. [Online; accessed May 30, 2019].
- Neumann, M.; Shen, Z.; and Skjonsberg, S. 2021. PAWLS: PDF Annotation With Labels and Structure.
- Roberts, K.; Alam, T.; Bedrick, S.; Demner-Fushman, D.; Lo, K.; Soboroff, I.; Voorhees, E.; Wang, L. L.; and Hersh, W. 2020. TREC-COVID: rationale and

structure of an information retrieval shared task for COVID-19. *Journal of the American Medical Informatics Association* : JAMIA 27:1431 -- 1436.

Shen, Z.; Zhang, R.; Dell, M.; Lee, B. C. G.; Carlson, J.; and Li, W. 2021. LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis.

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

Subramanian, S.; King, D.; Downey, D.; and Feldman, S. 2021. S2AND: A Benchmark and Evaluation System for Author Name Disambiguation.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, 697--706. New York, NY, USA: ACM.

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

Vrandečić, D., and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57(10):78--85.

Wang, L. L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; Yang, J.; Burdick, D.; Eide, D.; Funk, K.; Katsis, Y.; Kinney, R. M.; Li, Y.; Liu, Z.; Merrill, W.; Mooney, P.; Murdick, D. A.; Rishi, D.; Sheehan, J.; Shen, Z.; Stilson, B.; Wade, A. D.; Wang, K.; Wang, N. X. R.; Wilhelm, C.; Xie, B.; Raymond, D. M.; Weld, D. S.; Etzioni, O.; and Kohlmeier, S. 2020. CORD-19: The COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. *LayoutLM: Pre-Training of Text and Layout for Document Image Understanding*. New York, NY, USA: Association for Computing Machinery. 1192–1200.

Zeng, A.; Sabek, I.; and Cafarella, M. 2021. Unpublished analysis of Wikidata dumps.

Zhang, Y.; Zhang, F.; Yao, P.; and Tang, J. 2018. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, 1002–1011. New York, NY, USA: Association for Computing Machinery.

Michael Cafarella is a Principal Research Scientist at MIT CSAIL.

Michael Anderson is a research scientist in Computer Science at the University of Michigan.

Iz Beltagy is a Senior Research Scientist at the Allen Institute for Artificial Intelligence.

Arie Cattan is a graduate student in Computer Science at Bar-Ilan University.

Sarah Chasins is an assistant professor of Computer Science at the University of California, Berkeley

Ido Dagan is a professor of Computer Science at Bar-Ilan University.

Doug Downey is a Research Manager at the Allen Institute for Artificial Intelligence and a Professor of Computer Science at Northwestern University.

Oren Etzioni is Chief Executive Officer at the Allen Institute for Artificial Intelligence and Professor Emeritus, University of Washington.

Sergey Feldman is Senior Applied Research Scientist at the Allen Institute for Artificial Intelligence.

Tian Gao is a graduate student in Computer Science at the University of Michigan.

Tom Hope is a Postdoctoral Investigator at the Allen Institute for Artificial Intelligence.

Kexin Huang is a graduate student in Computer Science at the University of Chicago.

Sophie Johnson was a data science analyst at the Allen Institute for Artificial Intelligence.

Daniel King was an Applied Research Scientist at the Allen Institute for Artificial Intelligence.

Kyle Lo is a Senior Applied Research Scientist at the Allen Institute for Artificial Intelligence.

Yuze Lou is a graduate student in Computer Science at the University of Michigan.

Matthew Shapiro is the Lawrence R. Klein Collegiate Professor of Economics at the University of Michigan.

Dinghao Shen is a graduate student in Computer Science at the University of Michigan.

Shivashankar Subramanian was a graduate student in the Department of CIS at the University of Melbourne

Lucy Lu Wang is a Postdoctoral Investigator at the Allen Institute for Artificial Intelligence.

Yuning Wang was a student in Computer Science at the University of Michigan.

Yitong Wang is a graduate student in Computer Science at the University of Michigan.

Daniel S. Weld is Chief Scientist of the Semantic Scholar project at the Allen Institute for Artificial Intelligence and Professor Emeritus, University of Washington.

Jenny Vo-Phamhi is an Ertegun Scholar at Oxford University.

Anna Zeng is a graduate student in Computer Science at MIT.

Jiayun Zou is a graduate student in Computer Science at the University of Michigan.