Google Dataset Search: Building a search engine for datasets in an open Web ecosystem

Natasha Noy
noy@google.com
Google AI
Mountain View, California

Matthew Burgess
mattburg@google.com
Google AI
Mountain View, California

Dan Brickley
danbri@google.com
Google
Mountain View, California

ABSTRACT

There are thousands of data repositories on the Web, providing access to millions of datasets. National and regional governments, scientific publishers and consortia, commercial data providers, and others publish data for fields ranging from social science to life science to high-energy physics to climate science and more. Access to this data is critical to facilitating reproducibility of research results, enabling scientists to build on others' work, and providing data journalists easier access to information and its provenance. In this paper, we discuss Google Dataset Search, a dataset-discovery tool that provides search capabilities over potentially all datasets published on the Web. The approach relies on an open ecosystem, where dataset owners and providers publish semantically enhanced metadata on their own sites. We then aggregate, normalize, and reconcile this metadata, providing a search engine that lets users find datasets in the "long tail" of the Web. In this paper, we discuss both social and technical challenges in building this type of tool, and the lessons that we learned from this experience.

KEYWORDS

data discovery, search, metadata, structured data

ACM Reference Format:
Natasha Noy, Matthew Burgess, and Dan Brickley. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *Proceedings of The Web Conference (WebConf'2019).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313685

1 WHY IT CAN BE DIFFICULT TO FIND DATASETS ON THE WEB

Data is the main substrate of research for scientists in many disciplines, for the work of journalists, for the analysis by policy makers, and for those of us who are curious to understand our world better. Data is published on the Web by national and regional governments, scientific publishers, commercial data providers, research consortia, specialized data repositories, data aggregators, and so on. There are thousands of data repositories on the Web [16] and many individual data publishers, from high school teams publishing the results of their science-fair projects to national research labs. The more data we publish on the Web, the more complex the problem of data discovery becomes: how do we find the data that we need for the task at hand and how do we assess its credibility, veracity, and suitability for our task? The following are some of the key factors that contribute to the difficulty of data discovery today.

Proliferation of data publishers: In the early days of the Web, many users discovered what was available on the Web by browsing through directories such as the Yahoo! Web directory. Eventually though, the Web became too large for any directory to be sufficiently comprehensive and *search* became the primary way of finding where the information was on the Web. Today, the world of publishing datasets on the Web is in a similar transition: there are a number of well respected directories of dataset publishers (e.g., DataCite [29], re3data [16], Scientific Data in Nature [19]), but now they inevitably miss new datasets or repositories [7]. In addition, they miss datasets published by individual data providers, such as a scientist publishing the dataset resulting from her experiment on her own site. Finally, many curated resources collecting dataset repositories focus largely on government and research data and miss repositories that come from the private sector. Yet, there are valuable datasets that originate from companies who may charge or have more restrictive licensing terms for their data (e.g., ceicdata.com). Thus, the time is ripe to add Web-wide search capabilities for datasets and not to rely exclusively on curated directories.

The "long tail" and the "deep Web": Web search engines often fail at finding data for a variety of reasons: many pages describing datasets are in the "long tail" of the Web [8]. In many cases, data repositories do not make the individual pages for datasets easily crawlable because they are accessible only through queries, a phenomena referred to as the "deep Web" [18]. Consider, for example, Open Data Network [23], which hosts many datasets from local governments in the United States. It is a resource that many data journalists use (but many social-science researchers may not know about). The only way to get to a page for any specific dataset that Open Data Network hosts is by typing in a search term where that dataset will be listed among the results. For many such pages, search engines end up indexing some datasets as they appear in search results and not others.

Specialization in data-publishing communities: Research has become more interdisciplinary and scientists must be able to find data in the disciplines that they are less familiar with. A professional will most likely know her go-to data repositories in her specialized field by reading relevant papers, by asking colleagues, and through word of mouth. Then as she looks for data, she most likely uses the search tools within a specific data repository, which are usually perfectly suited for that type of data (e.g., social-sciences data, environmental, economic). However, these methods no longer work when you start looking for data in a discipline that you are

WebConf'2019, May 2019, San Francisco, CA USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/3308558.3313685

not intimately familiar with. For example, an epidemiologist trying to understand how a new virus spreads may want to understand environmental factors that coincided with the spread of the virus, yet she might not have any colleagues who know the landscape of the environmental data well. If you are a new researcher, or someone who is not very "plugged into" a research community, you might not even have anybody to ask. Just as Google Scholar has revolutionized vertical search for academic publishing, a specialized vertical search engine can fundamentally improve data discovery across all scientific disciplines.

In this paper, we discuss Google Dataset Search (https://g.co/datasetsearch), a search engine over dataset metadata that we built with an open ecosystem at its core: data publishers, large and small, use Schema.org or W3C DCAT markup to describe the metadata semantics on the individual pages of each dataset, a Web crawler collects this metadata, and Dataset Search augments and indexes this metadata to provide a vertical search over all the dataset metadata on the Web. As new publishers add metadata markup to their sites, Dataset Search includes them after the crawler visits and processes the pages. Dataset Search was launched as public beta at the beginning of September 2018.

Specifically, this paper makes the following contributions:

- We **present a search engine over dataset metadata** that can scale to all metadata published on the Web through an open ecosystem.
- We identify a set of **challenges** in building a search engine over structured data in such an open ecosystem.
- We describe our **technical approach** to normalize metadata, to reconcile it to a knowledge graph, to identify duplicates, and to index the data to provide a data-discovery system in the context of the whole Web.
- We discuss practical **lessons learned** from building a vertical search engine in an open ecosystem.

2 RELATED WORK

There are several key lines of work that both informed and enabled our work on Dataset Search.

Many scientific disciplines have come to a consensus (or have been compelled by funding agencies and academic publishers) that it is important to publish data and to cite it. There are, for example, peer-reviewed journals dedicated to publishing valuable datasets, such as Nature Scientific Data [19]. There are efforts such as DataCite [29], which provide digital object identifiers (DOIs) for datasets and both encourage and enable scientists to publish their datasets. Providing mechanisms for citing data enables scientists to get credit for publishing the data. Many scientific communities have signed on to the principles of management and stewardship of the data in a FAIR way (findable, accessible, interoperable, and reusable) [34]. Recently there have been efforts to create standards around citation of datasets. Fenner and colleagues describe a roadmap for implementing the Joint Declaration of Data Citation Principles [7]. Among the principles outlined in the article is the requirement that datasets have a landing page and a recommendation that the landing pages use Schema.org to describe its contents.

Naturally, as scientists pay more attention to publishing data, many scientific communities create data portals for scientists in that community to publish their datasets. These portals are usually designed for the specific type of data in that domain, and focus on the interface and interaction that suits scientists in that domain. There are hundreds of such repositories, with re3data.org, a registry of research data repositories, listing more than 2,000 repositories.

It is no wonder that such proliferation of repositories leads to the problem of data discovery. A qualitative study by Koesten and colleagues [17] found that more than 50% of users who searched for data using a mix of Web search engines, domain-specific repositories, and recommendations from colleagues reported difficulty in finding the datasets that they need. Kacprzak and colleagues [13] analyze the differences between search behavior for datasets and Web searches. Their analysis of query logs from four major open data portals, showed that queries for datasets are short, portal search engines are used in an exploratory manner and that the topics of queries for data differ from topics in Web search queries. Their findings indicate that the tools for querying datasets are not as well developed as Web search engines, and require more human work (e.g., scrolling through more search results returned from short queries) than Web search.

One approach for providing access to multiple data repositories is to harvest the metadata from these repositories through their APIs. For example Open Data Portal Watch [20] collects metadata from more than 260 repositories, focusing on government data. The Open Data Portal Watch developers evaluate and compare the quality of the metadata for these repositories. DataMed [22] is an example of a similar approach in the domain of life sciences, collecting metadata from 75 repositories. These approaches inherently rely on having the portal owners know the list of dataset repositories in advance.

A complementary approach to collecting metadata from a specific set of dataset publishers is to allow dataset publishers to submit their datasets to the repositories. Data-management systems such as CKAN [3], Quandl [26], Kaggle [14], and Microsoft Azure Marketplace [1] are repositories of data from multiple sources, organized for distribution and sharing. In all these systems, dataset owners actively choose to contribute their datasets to the system or to annotate the datasets with metadata. However, in all these systems, if dataset owners want their dataset metadata to appear in more than one repository, they need to submit the description of their dataset in each of those repositories separately.

The distinguishing characteristic of Dataset Search is the open ecosystem: the owners of datasets or data repositories need to markup the metadata only once, in a place where they publish and maintain it on their own site. Once they do, any Web crawler can pick it up and the datasets from this new source will show up in Dataset Search, or in other tools that rely on this metadata. In other words, repository owners do not need to submit their datasets anywhere proactively or to have a data portal know about the existence of their repository. As long as their Web pages are accessible to a general Web crawler and have the structured metadata on the pages, their datasets will be included. Researchers have explored this "post-hoc" approach in the context of enterprise data management [12]. To the best of our knowledge, Dataset Search is the first such approach for datasets in the context of the entire Web.

We chose to rely primarily on Schema.org for describing dataset metadata because both search engines and open-source tools have used it successfully to build an open ecosystem for various types

of content [11]. Erickson and colleagues [5] first proposed to use Schema.org to describe open government data and provided the first definitions for `http://schema.org/Dataset` based on W3C DCAT [4]. And in recent years, scientific community has also embraced it for publishing data: There are mappings from other metadata standards to Schema.org. For example, Sansone and colleagues define a mapping from the DATS standard in the biomedical community to Schema.org [30]. Wang and colleagues use Schema.org to describe research-graph data, which captures researchers, datasets and scholarly articles [33]. Efforts such as bioschemas.org [10] extend Schema.org to include domain-specific terminology and relationships. All these efforts enabled Dataset Search to start with an ecosystem that already had some metadata in Schema.org. Our team could then work with these communities to build on the existing efforts to encourage adoption.

3 DEFINING THE PROBLEM

In order to define the problem of dataset search on the Web and to highlight the key requirements for a product that would support it, we conducted interviews with scientists, journalists, and other data consumers. Many of them emphasized that outside of any niche community, nobody can enumerate all the dataset repositories that exist, even in a single discipline. Therefore, our goal was to enable an open ecosystem that anybody can join at any time: from a large institutional repository to a group of students who want to publish the results of an experiment.

When defining the problem of building a search engine over datasets, the first definition we must consider is the definition of a dataset, and, more specifically, a **dataset as published on the Web**. For the purposes of the work described in this paper, we have adopted an operational rather than a declarative definition: anything that a data provider considers to be a dataset is a dataset. This definition means that we can include individual tables, files, images, binary files, maps, or collections of any of the above. On the one hand, this definition removes the need to "police" what a dataset is and allows providers in different domains to share data in a format that makes sense to them (e.g., binary NetCDF for climate data). On the other hand, this approach puts a bigger burden on tools such as Dataset Search to ensure that its users get high quality and meaningful results.

Furthermore, we focus on the problem of search over dataset **metadata**, which includes data *about* a dataset. Metadata describes salient properties of a dataset, such as its title and description, provider, spatial and temporal coverage, and so on (see http://schema.org/Dataset for a sample of metadata properties).

We define the task of **searching for datasets on the Web** in the following way: Given a set of Web pages that publish dataset metadata, unknown in advance, build a search engine over this metadata to enable users to find datasets on those pages. Specifically, this problem setting highlights the following requirements:

- The system needs to be open, allowing any new provider to join by publishing their metadata.
- The search is over metadata, and does not have to include the data itself. Indeed, a data provider may require the users to pay for data or to create a free account to obtain access (the licensing terms themselves are part of the metadata).
- The metadata must be published by the dataset publishers themselves, using a standard that our and other solutions can interpret.

4 TECHNICAL CHALLENGES

We now describe the technical challenges that we had to address in building Dataset Search. Most of these challenges would apply to building any large-scale vertical search engine in an open ecosystem, and are not specific to datasets.

4.1 Metadata quality

At its core, Dataset Search is a search engine over metadata provided by data publishers on the Web. As a result, the quality of the metadata varies greatly. We have observed that *"everything that can go wrong, will go wrong"* when you are operating at this scale in an open ecosystem. For a simple example, consider representation of dates, which appear in several properties of dataset metadata, including publication date of a dataset, its modification date, and its temporal coverage. The Schema.org standard requires that dates follow the ISO 8601 format (e.g., `"2018-11-01"`). However, we have found dates in every possible format, including sentences such as `"Published in December, 2015"`—in different languages. Similarly, spatial-coverage definitions in metadata may include latitude and longitude for points defining a geographic shape. We have observed cases were providers mixed up the order of latitude and longitude. In some cases, dataset title and its identifier were swapped, and so on. Because metadata is usually generated programmatically, such problems usually affect all metadata from a given repository, which may include hundreds or thousands of datasets.

The Schema.org specification also leaves some details about how the providers should specify the metadata open to interpretation. For instance, one can specify `encodingFormat` either at the level of a dataset, or at the level of `distribution`, which is a property of a `http://schema.org/Dataset`. In the context of dataset publishing, the distinctions between publisher and provider are often blurred and imprecise. Other such examples abound. Thus, to make the search useful, we must clean up and normalize the metadata as much as we can (Section 5.2).

4.2 Metadata duplication in search results

In addition to the quality of individual metadata, the pages where metadata appears can also cause problems. Many dataset repositories are themselves search engines over the metadata for the datasets that they host. Therefore, a description of each dataset appears in two contexts in the repository pages: (1) in pages listing search results, and (2) on dataset profile pages for each individual dataset. We refer of these descriptions within the same site as **duplicates**. The potential number of pages in the first category is exponential, whereas the second category has one page for each dataset. Both types of pages are usually generated programmatically from a database or an index. We found that frequently developers attach metadata to each dataset in search-result listing, and may or may not attach it to the profile page. As a result, a crawler may potentially pick up an exponentially large number of copies of metadata for the same dataset, one from each search-result page where that dataset appears. However, from the point of view of a

dataset-search tool, the page that we really want to take the user to is the profile page for a dataset, and not a listing of search results. Thus, we must distinguish between these different types of pages and identify which metadata copy refers to the profile page (Section 5.4).

4.3 Dataset replication and provenance

While having multiple copies of metadata descriptions for the same dataset within a single repository is undesirable (Section 4.2), replication of the description of the same dataset in *different* repositories not only happens frequently but also provides a signal about the quality of the dataset itself. The presence of a dataset in multiple repositories often signals that the dataset is widely used and provides some indication about its credibility. We refer to the copies of metadata descriptions of the same dataset in different repositories as **replicas**. If we can cluster the replicas present in different repositories, we can give users a choice of which repository to go to. Hence, we face the challenge of identifying the replicas across repositories. The Schema.org standard provides a way to specify the original dataset explicitly through a `http://schema.org/sameAs` link: a dataset description can point to the original dataset description on another site through this link. Relying on `sameAs` links offers only a small part of the solution: First, the links are not always reliable. Second, when we examine the metadata that we collected, we find that less than 1% of dataset descriptions have explicit `sameAs` links to other domains, whereas we identified that about 25% of them are likely replicas of another dataset. Thus, we must rely on heuristics and other properties that could serve as proxies for `sameAs` as we cluster the replicas (Section 5.4).

4.4 Churn and stale sites

Search engines optimize their crawl schedules to crawl sites that their users visit often at a higher rate than the sites in the long tail [8]. While this optimization is entirely reasonable at Web scale, we need to tune it to improve the discoverability of content from the long tail by taking into account the utility of the metadata. Many of the pages that describe individual datasets are in that heavy long tail of the Web and therefore the problem of churn and stale links is exacerbated in this vertical. Even though we do not run our own crawl and rely on a general Google crawl, we still should not be sending users to stale pages. For the pages that we ultimately identify as dataset pages, we work with the crawl-scheduling team to ensure that we minimize the number of stale links.

While we cannot measure the churn among the datasets that the crawler does not see, we measure the churn in the dataset metadata for the descriptions in our index. For Dataset Search, over the past two months, on any given day, on average, 3% of the datasets are deleted from our index, and 7-10% of new datasets are added. We discuss our approach to addressing this churn efficiently in Section 5.5.

4.5 Ranking and relevance

One of the key research questions in building any vertical search engine is understanding what ranking signals make sense for that vertical. Initial studies indicate that there is a difference in how users expect datasets to be ranked [12, 15]. Because explicit links between dataset metadata are still relatively rare, traditional Web-based ranking methods are not effective. While we hope that eventually citing and referencing dataset will become as common a practice as citing and referencing scientific papers, it is not the case today. Thus, traditional scholarly metrics are not readily applicable either.

In addition, because metadata is often limited and minimalistic, it may not provide enough signal to decide whether a dataset is relevant to the user's query. We can fall back on the information on the rest of the page, if it is more expansive than the metadata, or recover additional metadata from the data itself (Section 5.6).

4.6 Multiple dataset-metadata standards

While Schema.org is widely used by search engines and other applications to improve many Web-based tools that need to rely on semantics of the data on a Web page, it is not the only open standard for describing dataset metadata. Several other standards exist, most notably, the W3C Data Catalog Vocabulary (DCAT) [4]. Mappings between Schema.org and the various extensions to DCAT are currently under discussion at W3C and Schema.org. We found that at the moment, only 2% of dataset descriptions (in JSON-LD, RDFa or Microdata) use the DCAT standard while the rest use Schema.org. However, the datasets that use the DCAT standard include hundreds of thousands of datasets from government portals around the world, and in particular portals with geo-spatial data. Therefore, to get better coverage and to be inclusive of other standards, we process both Schema.org and DCAT metadata, as long as the latter is also represented syntactically in a supported syntax, to allow the regular crawl processing to extract the triples (Section 5.7).

5 IMPLEMENTING DATASET SEARCH

To enable the open ecosystem, we rely primarily on Schema.org, a non-proprietary format to describe metadata on the Web in a structured form, providing basic semantics (Figure 1). As we mentioned in Section 4.6, we also support DCAT. Any dataset provider can add simple markup to Web pages that describe datasets. Our Web crawler then crawls this page and parses the HTML and the embedded markup (either as RDFa, microdata, or JSON-LD). Dataset Search does not require its own crawl; it builds upon the crawl and extraction infrastructure of the general Google Web crawl. Specifically, we rely on the Google Web crawl, which already processes structured-data markup and generates triples for each page, regardless of the specific vocabulary that the providers use (Schema.org, DCAT, etc.). The crawl and extraction of the triples from the pages are not contributions of this paper.

Dataset Search collects the metadata, links it with other resources, and builds an index of this enriched corpus of metadata. Once we built the index, we can start answering user queries—and figuring out which results best correspond to the query.

We will now describe the details of the key components of the pipeline in Figure 1.

5.1 From Schema.org to protocol buffers

Google Web crawl and its associated processing, relying on the underlying standards, parses RDFa, Microdata and JSON-LD to a common graph data model, broadly corresponding to W3C's RDF

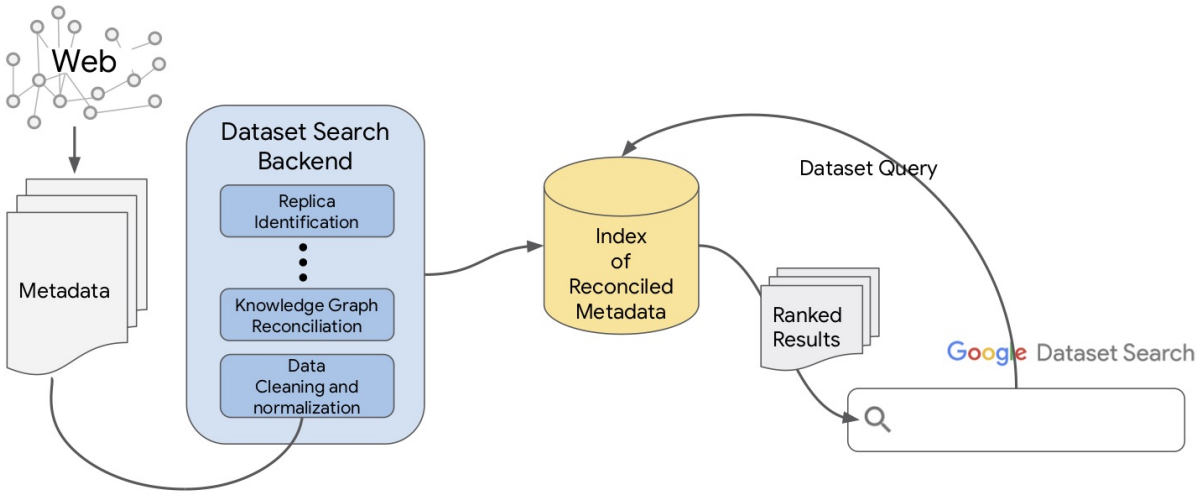

Figure 1: An overview of the Dataset Search components. Google crawler collects the metadata from the Web; Dataset Search backend normalizes and reconciles the metadata; we then index the reconciled metadata and rank results for user queries.

triples [28]. We then look for the triples that use our vocabularies of interest, Schema.org and DCAT. Specifically, we collect all the triples for all the pages that have elements of specific types: `http://schema.org/Dataset`, `http://schema.org/DataCatalog`, and `http://www.w3.org/ns/dcat#Dataset`.

For a set of triples from each page, we traverse the graph to collect all the properties and related objects for each dataset in a protocol buffer [32], a nested-relational record corresponding to each metadata entry. A dataset record can point to other records such as organizations that provided a dataset or a record describing the distribution of a dataset. A single Web page can have multiple dataset records on it.

The specification of the graph traversal captures the mapping from Schema.org and DCAT vocabularies to the corresponding elements in the protocol-buffer definition (e.g., example fields in Figure 2). The schema of the protocol buffer for the metadata largely corresponds to `http://schema.org/Dataset` and therefore the transformation of metadata at this stage is rather small.

To improve scalability, we use the graph query independently on the triples from each individual page rather than try to extract information from a graph that includes all metadata triples on the Web. Because the links across different pages must specify objects on another page directly through a URL (e.g., a provider of this dataset on page *A* is described on page *B*), we can do this reconciliation post-hoc. So, essentially, each page corresponds to its own, possibly disconnected graph. At the same time, doing graph traversal only for a single page is dramatically more scalable.

The information that we extract through graph traversal constitutes the **raw metadata**, metadata that closely mimics the structure of Schema.org properties on the original page.

In the next few steps, we describe how we create **reconciled metadata** for each dataset, accounting for the different levels of quality and variety of the modeling patterns used.

5.2 Normalizing and cleaning the metadata

As we mentioned in Section 4.1, we must assume that we will encounter every possible misuse and mis-interpretation of Schema.org properties when we operate at the scale of the whole Web. Thus, we perform a number of operations to normalize and clean up the metadata.

First, for the properties where we observe different patterns on the Web, we analyze the common patterns used and try to account for all of them. For instance Figure 2 shows the different patterns that we observed for defining downloads and distribution. In the figure, the first example of raw metadata defines the format of the dataset (CSV) at the level of the dataset itself and stores the download URL as the value of the `http://schema.org/distribution` property. Other examples in the figure deal with these two pieces of information differently. All these patterns are commonly used in our corpus. We mine these patterns by traversing either the initial graph or the resulting protocol buffer. Once we identify the patterns, we write adapters to convert all of them into the same modeling pattern in the reconciled metadata record. The right-hand side of Figure 2 shows this reconciled result.

Similarly, we have developed adapters for other metadata fields: We understand a lot more representations of dates than the ISO standard required by the Schema.org specification (Section 4.1. We will pick up digital object identifiers (DOIs) for a dataset from a variety of fields, and not just `http://schema.org/identifier`. We will use a uniform field, `provider`, for the many different fields that dataset providers used to identify this property. As we collect more metadata, our set of such adapters grows. Our decisions in these steps are guided by two factors: (1) the frequent usage patterns that we observed in the data; and (2) our understanding of what we expect the users to see in Dataset Search results.

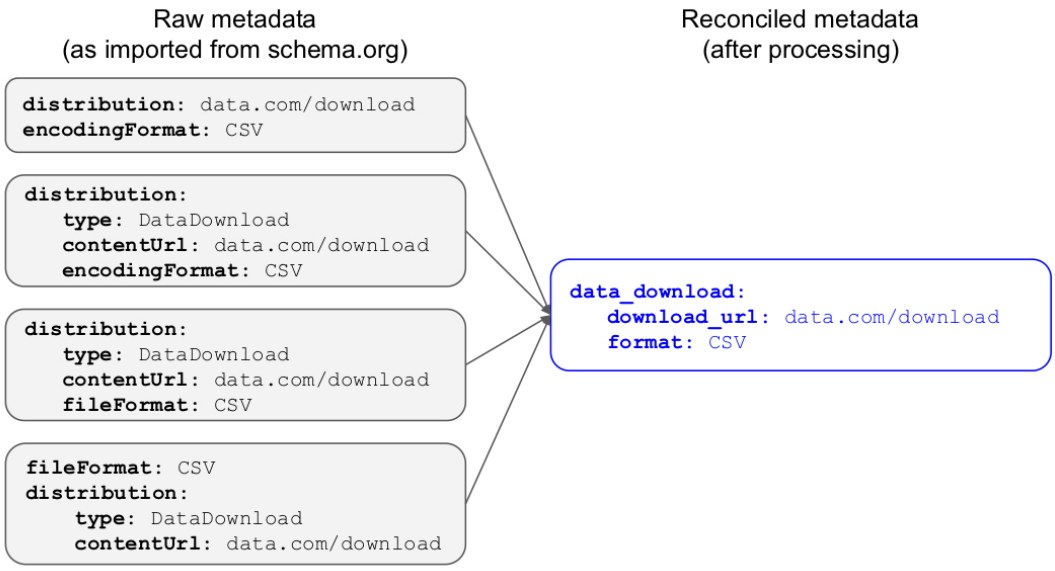

Figure 2: Normalizing raw patterns. The diagram gives an example of the variety that we have observed in metadata definitions. The examples on the left are a protocol-buffer version of the representation, as imported directly from Schema.org metadata. All these examples reconcile to the same pattern, shown on the right.

5.3 Reconciling metadata to a knowledge graph

Most companies today have a knowledge graph that describes types of entities and relationships between them that are critical to the company's products. In our case, such a knowledge graph describes and links information about many entities, including the ones that appear in dataset metadata: organizations providing datasets, locations for spatial coverage of the data, funding agencies, and so on. Therefore, we try to reconcile information mentioned in the metadata fields with the items in the knowledge graph. We can do this reconciliation with good precision for two main reasons. First, we know the types of items in the knowledge graph and the types of entities that we expect in the metadata fields. Therefore, we can limit the types of entities from the knowledge graph that we match with values for a particular metadata field. For example, a provider of a dataset should match with an organization entity in the knowledge graph and not with, say, a location. Second, the context of the Web page itself helps reduce the number of choices, which is particularly useful for distinguishing between organizations that share the same acronym. For example, the acronym CAMRA can stand for "Chilbolton Advanced Meteorological Radar" or "Campaign for Real Ale." If we use terms from the Web page, we can then more easily determine that CAMRA is in fact the Chilbolton Radar when we see terms such as *clouds*, *vapor*, and *water* on the page.

This type of reconciliation opens up lots of possibilities to improve the search experience for users. For instance, Dataset Search localizes results by showing reconciled values of metadata in the same language as the rest of the page. Additionally, it can rely on synonyms, correct misspellings, expand acronyms, or use other relations in the knowledge graph for query expansion.

5.4 Finding duplicates and aggregating replicas

As we discussed in Sections 4.2 and 4.3, the same dataset description may appear in multiple places. We refer to such dataset descriptions as *duplicates* is they occur within the same site. Duplicates usually appear because site developers chose to mark up dataset descriptions both when listing search results and in profile pages for individual datasets. For Dataset Search users we should be indexing only one of the many duplicates within a single site, preferably the one that appears on the profile pages.

When a dataset (e.g., a popular one), appears in different repositories, we refer to these descriptions as *replicas*. Because we have somewhat of a "bird's eye" view of the Web, we can often find where all these different replicas are, and group them together, giving the user a choice of which repository they want to get this dataset from.

While presenting multiple duplicates is not useful for users, replicas from different repositories can be useful for a user, giving her a choice to go to the repository that she trusts or knows.

Our approach to identifying duplicates and replicas are very similar—with the main difference being whether or not we include the site domain into consideration.

First, Schema.org has a way to specify the connection explicitly, through `http://schema.org/sameAs`. In our experience, when this property was present, it was a strong indicator that datasets are the same, whether they are duplicates within the same site or replicas across different sites. Indeed, if developers must include Schema.org for metadata in search results in addition to the profile pages on their site, our guidelines[1] suggest using `sameAs` to point to the profile page.

[1] https://developers.google.com/search/docs/data-types/dataset

Second, we found that sometimes two datasets can be identified as duplicates or replicas not because they point to each other through `sameAs`, but rather because they both point to the same canonical page for that dataset. The latter page itself might not even have Schema.org metadata and hence will not be in our corpus.

Finally, we use a number of other heuristics that, in combination, provide a signal on possible duplicates or replicas: for example, two datasets sharing a large part of their metadata; or pointing to the same URL as their download URL, are likely related. In practice, each one of these signals can fail or be subject to mis-representation or misinterpretation by data providers. So, we use a combination of signals to determine whether or not dataset descriptions appear to be the same.

To identify these duplicates and replicas at scale, we compute a hash value (fingerprint) for each combination of values for a dataset that may be an indication that it is a replica or a duplicate of another dataset. For instance, these combinations of values may include dataset title and description, or dataset title and a download URL. We heavily pre-process the fields that we use to compute the fingerprint: we normalize the text, remove delimiters and any special symbols, and remove spaces. This normalization is usually sufficient because data providers normally copy the metadata directly from one site to another. Once we have the fingerprints for these field combinations, we construct a graph where nodes represent datasets and edges connect nodes that share at least one fingerprint (Figure 3). Our goal is to identify connected components within this graph: each connected component is a cluster of duplicates, if the nodes are from the same site, or replicas, if they are from different sites. In the example in Figure 3, datasets *A*, *B*, and *C* are all duplicates (or replicas). Even though datasets *A* and *B* do not share any fingerprints; they are both the same as dataset *C*. Hence, we can infer that they are part of the same cluster.

We first use this algorithm to compute duplicates, grouping only datasets within a single domain by adding the domain to each fingerprint, thus ensuring that descriptions from different domains will never end up in the same cluster. We then choose one representative from a cluster (Figure 4). We use heuristics to maximize the probability that this representative refers to a profile page for a dataset and not a listing of search results. We then run the same algorithm to compute replicas on this de-duped set. We use a MapReduce version of a connected-component algorithm [27] to compute these clusters efficiently.

Table 1 shows the distribution for different sizes of *replica clusters*. Recall that replicas are cases where the same dataset appears on different sites. The vast majority of clusters have only two replicas, whereas there are a few with as many as 40 or 50 replicas in different domains, with the largest cluster having 58 datasets.

5.5 Scalability of the backend

As with any system that operates at Web scale, we must address scalability of our implementation. We discuss the scalability of the Dataset Search backend from Figure 1, because we could rely on other, already scalable, systems for crawl, indexing, and querying.

Figure 4 highlights the backend architecture. After we traverse the graph of triples on each individual page, we generate a protocol buffer (a record) for each dataset. We then do most of the processing

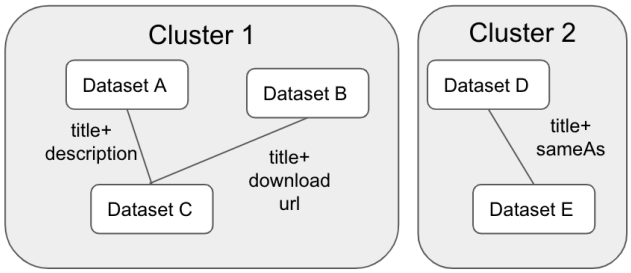

Figure 3: Clusters of replicas or duplicates as connected components. Each pair of datasets may share some fingerprints (e.g., based on title+description). We identify connected components in the graph to cluster all replicas of the same dataset. Table 1 profiles the sizes of clusters that we found.

Table 1: Distribution of replica clusters. Each cluster groups together dataset descriptions from different Web domains that we determine to be replicas.

Number of datasets in a replica cluster	Fraction of clusters
2	82.57%
3	15.17%
4	1.88%
5	0.28%
6 and larger	0.11%

that we described in earlier sections (cleaning, normalization, reconciliation to the knowledge graph) on each dataset independently and in parallel, using MapReduce. Note that for each enrichment that we discuss, we do not need to know how the other records look like. Furthermore, because only a small percentage of datasets changes from day to day (Section 4.4), we do not need to re-process the majority of the datasets, and only re-process the new or modified ones. This architecture is inspired by enterprise dataset search, such as Goods [12].

The identification of duplicates and replicas, however, must be done for the entire corpus each time, because newly added datasets may appear in any of the clusters. Thus, we run the clustering algorithms for identifying both replicas and duplicates daily.

5.6 Indexing and ranking the results

The collection of structured metadata and the replica-aggregation results go into our index. For each dataset, we index the links to its replicas along with its metadata to enable us to retrieve all the information about the whole cluster at the same time. When we collect the results returned from the index, we cluster the replicas together, to show them as a single entry in the list of results.

When the user issues a query, we search through our corpus of datasets using the same methodology as a Web search engine. Just like with any search, we need to determine whether a document is relevant for the query and then rank the documents in terms of how relevant and how important they are. Because there are no large scale studies on how users search for datasets, as a first

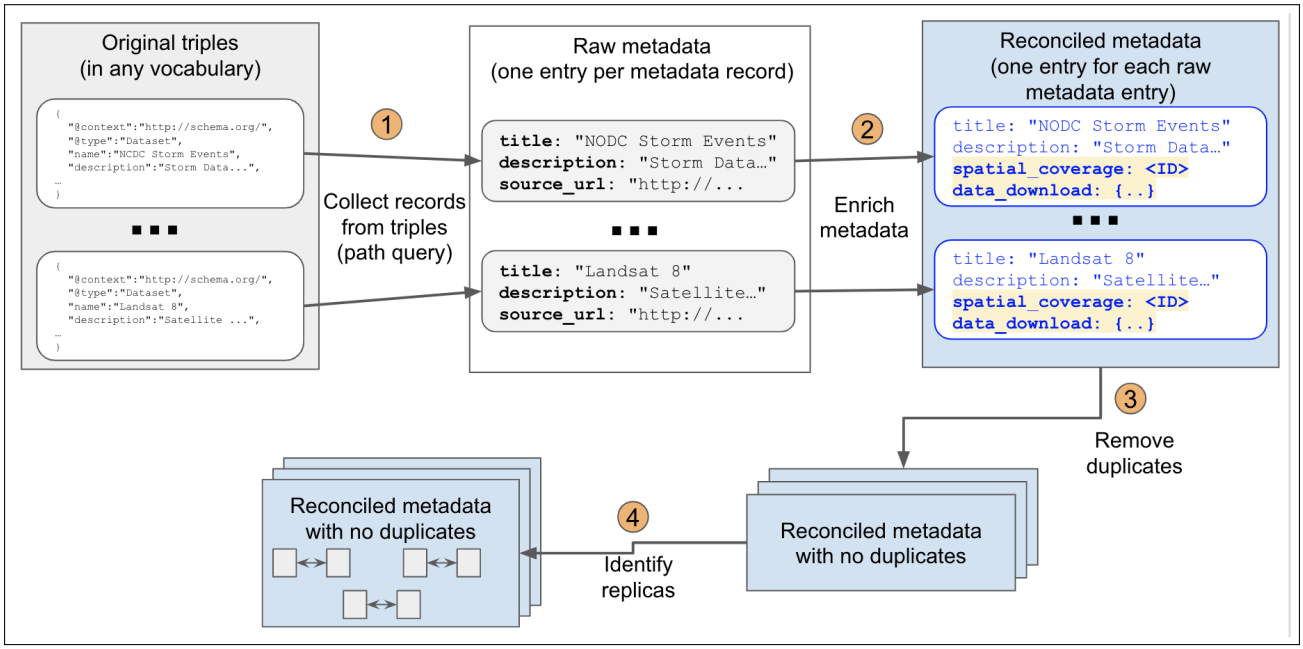

Figure 4: Dataset Search backend: Processing triples on each page results in individual records for each dataset. We then process each record independently to enrich the metadata (e.g., the highlighted portion for reconciled metadata on the right). Steps 1 and 2 happen in parallel, using MapReduce. The algorithms on finding duplicates (step 3) and replicas (step 4) then run on the whole corpus.

approximation, we rely on Web ranking: after all, metadata records come from Web pages. And while this approach provides a good starting point, it is not enough: because many of the pages are in the long tail, the differences in ranking are often not very meaningful. Thus, we add signals that take into account metadata quality. We hope that as we gain more insight on how users query and use Dataset Search, we will be able to develop ranking models that are more specialized for searching datasets.

Finally, Figure 5 shows a screenshot of the user interface: As the user enters a search strings, she gets a list of datasets, with some salient metadata, along with one or more domains where we found that dataset. The page with details for the selected result shows the reconciled metadata.

5.7 Improving coverage and recall

In Section 4, we highlighted challenges to having as broad coverage as possible. First, relying exclusively on the Schema.org vocabulary misses many government sites and we would like to be able to process triples that use other vocabularies or ontologies (Section 4.6). Recall that in the first step of our pipeline, we use a graph traversal to go from a set of triples corresponding to a page to a protocol buffer that represents the record for a dataset metadata. We have extended the graph traversal to understand the DCAT vocabulary as well, using the mapping between the two vocabularies [24]. The DCAT essentially becomes just another vocabulary that we must be able to understand. We referred to Schema.org throughout this paper, mostly to make it easier to read, note that we include the DCAT standard as well. Supporting multiple vocabularies requires some maintenance overhead to ensure that mappings are still valid, we do not expect the number of vocabularies to grow or for the mapping to change frequently. Because most of the tooling is still optimized for Schema.org, we usually still recommend to our partners to use Schema.org if it is feasible for them.

As we mentioned in Section 4.2, in some repositories, search is the only way to get to individual datasets. Thus, it may be hard for a crawler to get a full list of datasets in a repository. Sitemaps have been a common solution to this problem and we have strongly encouraged dataset-repository owners to maintain sitemaps that list all their datasets in our guidelines.[2]

6 THE STATUS OF DATASET METADATA ON THE WEB

Several researchers and practitioners have encouraged the use of Schema.org for dataset discovery on the Web in the last 1-2 years (e.g., [7]). Our team also participated in such efforts [21]. This community building was necessary prior to launching any vertical search based on this metadata: We needed to avoid the cold-start problem of not having any metadata to search over when a product is launched. Our approach involved public blog posts [21] and presentations to raise awareness of problems in data discovery today and the need for standardized Web-based metadata. We have also identified 2-3 disciplines for which we attempted to have more comprehensive coverage. We consulted experts in those disciplines to get a list of prominent repositories in those fields, the repositories

[2]https://developers.google.com/search/docs/data-types/dataset

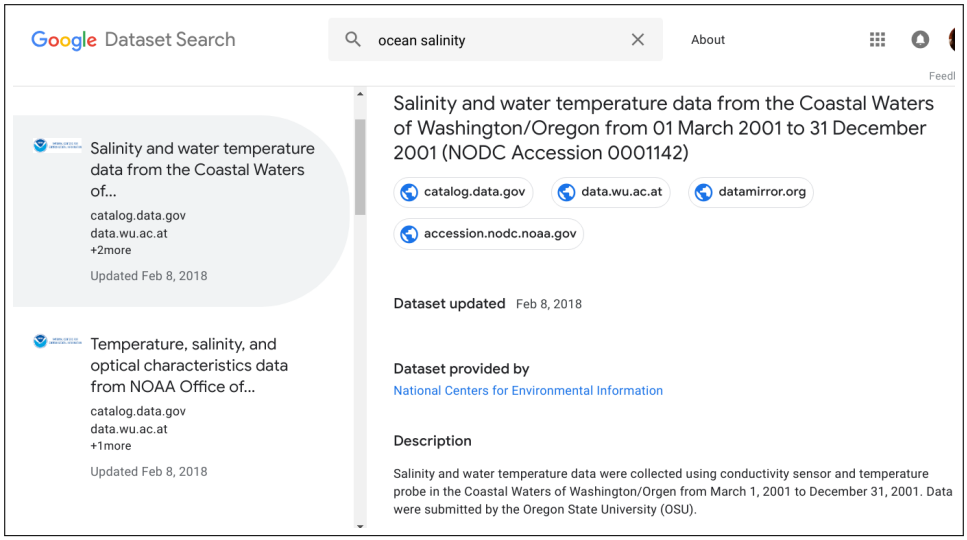

Figure 5: Dataset Search: Query results show the list of matches and the details for each match. The example in the figure shows multiple repositories where the metadata for the selected dataset is available.

without whom any dataset search is not viable. We then worked with the developers of these repositories directly to bootstrap the ecosystem. As the leaders in the domains added the metadata to their pages, developers of other repositories often followed suit.

We found that while having key partners gave us confidence that we will have datasets that many scientists consider important, the majority of domains by the time we launched were not from the partners that we explicitly contacted. At the time of the launch, there were thousands of domains that had `http://schema.org/Dataset` and millions of datasets. As the products that use this metadata begin to appear, the number of domains and datasets is growing: about 30% increase in the number of domains and 90% increase in the number of datasets in the first three months.

7 DISCUSSION

We now discuss the lessons that we learned when building Dataset Search (Section 7.1), the feedback that we received from users after the initial public beta launch (Section 7.2), and future work (Section 7.3).

7.1 Lessons learned

In our efforts to build Dataset Search based on an open ecosystem, we have learned several lessons that we think might be valuable to others building similar vertical search engines or those who work on tools and applications that rely specifically on dataset metadata.

Build an ecosystem first: Often, the immediate impulse of technologists is to build technical solutions for a given problem. In the case of Dataset Search, a key technical problem is to identify and extract metadata for datasets spread across the entire Web. Open information extraction from long tail sources is a known hard problem (e.g., [6, 25, 31]). Therefore, posing the problem of collecting metadata as a solely technical problem makes the solution significantly less feasible. Relying on an open source standard that enables dataset providers to describe their datasets and creating an ecosystem for publishing metadata led to several benefits. First, we have an immediate solution to the problem of collecting metadata at Web scale. This problem would have been a much more daunting and time consuming feat if we used a purely technical solution. Second, the majority of dataset repositories already store this metadata in a structured form in their own database, generating dataset profile pages programmatically from these databases. Rather than trying to reverse-engineer this metadata from a variety of fonts, layouts, frames, and so on, we get much better fidelity by simply having this structured data populate the corresponding metadata fields. Finally, we can think of the ecosystem as a means of bootstrapping a technical solution: The ecosystem provides an abundant supply of data needed to train systems that may be able to assist in identifying more landing pages with metadata or enriching the metadata we have already collected (Section 7.3).

Open non-proprietary standard is key: When deciding which standard to adopt for the metadata markup we chose non-proprietary open standards, Schema.org and DCAT. It takes about the same amount of development or manual effort on behalf of dataset owners or repository developers to create metadata that conforms to any standard, proprietary or otherwise. However, the value of markup that uses a non-proprietary standard potentially can be many times that of a proprietary standard because many different search engines and other applications can use the markup easily. We found that it was much easier to encourage developers to provide markup in non-proprietary standard than it would have been in a proprietary one: many commented that the perceived increased value because markup can be leveraged by multiple applications was important. Just as significant was the understanding that they were not adding the metadata to benefit a single corporation or a single search engine but rather enhancing the discoverability of their data across all tools.

Bootstrapping requires influencers and incentives: One of the hardest aspects of creating a thriving data ecosystem is the process of bootstrapping. It is a chicken-and-egg problem: We need data providers to use the markup and data providers need some meaningful application that uses that markup to justify implementing it. A couple of factors made this task easier for the case of building Dataset Search. First, the open-data community, which builds many of the dataset repositories, generally understands the value of metadata and explicit semantics in its description. For the most part, this audience is as receptive as one could find to these ideas. Second, the encouragement was coming from teams that were planning to deploy the solution at a major search-engine company, Specifically, Google has used Schema.org in many of its products [11]. This history gave the developers a sense of how this metadata may end up manifesting itself in such products. It is important to note, however, that influence does not have to be purely based on traffic or the number of users. The influence may come from funding agencies, for example. We believe that using a non-proprietary standard is necessary but not sufficient for a healthy ecosystem: until the ecosystem is sufficiently large, many data providers will still need some incentive to implement the markup.

Semantics and knowledge graph are critical ingredients: We use lightweight semantics in many parts of Dataset Search. The Schema.org standard itself allows data providers to specify semantics of the content of a Web page using agreed upon vocabulary. In Section 5.3, we discussed the use of semantics to improve the accuracy of reconciliation to the knowledge graph and the use of the knowledge graph itself to add semantics to search, similar to other semantic-search approaches [2]. Furthermore, the key ingredient of our work is the core idea underlying the Semantic Web: resources on the Web should be linked and have their semantics described using shared vocabularies.

7.2 What users want

After collecting user feedback in the first month after the launch, we have learned what features our users seem to desire the most. By far the most requested feature for Dataset Search was to include some sort of faceted browsing. While free text queries are easy to use, users also want to be able to filter the results by date, location, size of a dataset, licensing terms, and other attributes. One of the reasons why we did not include this feature initially was the sparsity of metadata. the majority of dataset descriptions simply lack any values for the fields that one would naturally include in filtering. Thus, filtering results by the values of these sparsely populated fields, will severely impact recall in a faceted search. We hope that as the ecosystem grows and tools like Dataset Search highlight the specific attributes that we use, metadata providers will find it useful to specify values for these attributes, thus making filtered search more feasible.

One of the major components in Dataset Search is the identification of replica datasets across different repositories (Section 5.4). This feature was well received and users have asked that we infer more than simple replica relationships: Not surprisingly, data provenance is key for many researchers who rely on the data [9]. If we can identify the primary or canonical source for the data, we can highlight it better, giving users the choice to go to that source, or to another repository that they are more familiar with.

Finally, users have also requested that we index the data itself in addition to metadata. There are many challenges associated with obtaining, parsing, and indexing the raw data but we hope to include this feature in future iterations of Dataset Search.

7.3 Future work

There are many technical challenges that we need to address to make Dataset Search a more comprehensive and useful tool for dataset discovery.

First, we need to improve the **quality of the metadata** that we have by learning from the existing metadata, by linking to other resources such as academic publications, by understanding data provenance and the evolution of the metadata on the Web. We also need to make sure that the metadata faithfully describes the content of the page: technically nothing stops a data provider from adding `http://schema.org/Dataset` to a page describing a job posting, for instance. And indeed, we have observed many such instances in the data that we crawled. We hope that now that there is a huge variety of dataset pages that are marked up with Schema.org, researchers can use these pages to learn how a page describing a dataset might look like and to build classifiers for such pages.

To improve the **ranking of datasets**, we need to learn from the user interactions. As with any ranking problem, the result users click on for a given query, can help us build new models to improve the ranking.

We hope to improve the **coverage** both by encouraging the growth of the explicit metadata on the Web and using existing metadata for training methods that can extract new metadata.

Finally, we hope that the presence of tools that treat datasets as prominent first-class objects and encourage/reward citing of the data, will lead to an ecosystem where data owners find it valuable and rewarding not only to publish their data but also to describe it better and more fully.

8 CONCLUSIONS AND FUTURE WORK

We have described Dataset Search, a vertical search engine for data discovery that is based on an open ecosystem where data providers describe their metadata in Schema.org, an open non-proprietary standard. Using this community standard has allowed us to grow the coverage quickly, before any applications or products that rely on this metadata even appeared. The response from users to the launch of Dataset search has been extremely positive, not only from scientists and data geeks, but also from journalists and government agencies. We hope that the number of dataset repositories that publish their metadata continues to grow and that other tools start using more actively the metadata that these repositories publish, building other applications that make datasets first-class citizens in scientific and public discourse.

ACKNOWLEDGEMENTS

We would like to thank Xiaomeng Ban, Lee Butler, Thomas Chen, Corinna Cortes, Kevin Espinoza, Archana Jain, Mike Jones, Kishore Papineni, Chris Sater, Gokhan Turhan, Shubin Zhao and Andi Vajda for their work on the project and all our partners, collaborators, and early adopters for their help.

REFERENCES

[1] Azure marketplace. http://datamarket.azure.com/browse/data.

[2] Baeza-Yates, R., Ciaramita, M., Mika, P., and Zaragoza, H. Towards semantic search. In *International Conference on Application of Natural Language to Information Systems* (2008), Springer, pp. 4–11.

[3] CKAN. http://ckan.org.

[4] Data Catalog Vocabulary (DCAT). https://www.w3.org/TR/vocab-dcat/.

[5] Erickson, J. S., Viswanathan, A., Shinavier, J., Shi, Y., and Hendler, J. A. Open government data: A data analytics approach. *IEEE Intelligent Systems 28*, 5 (2013), 19–23.

[6] Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. Open information extraction from the web. *Communications of the ACM 51*, 12 (2008), 68–74.

[7] Fenner, M., Crosas, M., Grethe, J., Kennedy, D., Hermjakob, H., Rocca-Serra, P., Durand, G., Berjon, R., Karcher, S., Martone, M., and Clark, T. A data citation roadmap for scholarly data repositories. *bioRxiv* (2017).

[8] Goel, S., Broder, A., Gabrilovich, E., and Pang, B. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 201–210.

[9] Goodman, A., Pepe, A., Blocker, A. W., Borgman, C. L., Cranmer, K., Crosas, M., Di Stefano, R., Gil, Y., Groth, P., Hedstrom, M., et al. Ten simple rules for the care and feeding of scientific data. *PLoS computational biology 10*, 4 (2014), e1003542.

[10] Gray, A. J., Goble, C. A., and Jimenez, R. Bioschemas: From potato salad to protein annotation. In *International Semantic Web Conference (Posters, Demos & Industry Tracks)* (2017).

[11] Guha, R. V., Brickley, D., and Macbeth, S. Schema.org: evolution of structured data on the web. *Communications of the ACM 59*, 2 (2016), 44–51.

[12] Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., and Whang, S. E. Goods: Organizing Google's datasets. In *Proceedings of the 2016 International Conference on Management of Data* (2016), ACM, pp. 795–806.

[13] Kacprzak, E., Koesten, L. M., Ibáñez, L.-D., Simperl, E., and Tennison, J. A query log analysis of dataset search. In *Web Engineering* (Cham, 2017), J. Cabot, R. De Virgilio, and R. Torlone, Eds., Springer International Publishing, pp. 429–436.

[14] Kaggle datasets. https://www.kaggle.com/datasets.

[15] Kern, D., and Mathiak, B. Are there any differences in data set retrieval compared to well-known literature retrieval? In *Research and Advanced Technology for Digital Libraries* (Cham, 2015), S. Kapidakis, C. Mazurek, and M. Werla, Eds., Springer International Publishing, pp. 197–208.

[16] Kindling, M., van de Sandt, S., Rücknagel, J., Schirmbacher, P., Pampel, H., Vierkant, P., Bertelmann, R., Kloska, G., Scholze, F., and Witt, M. The landscape of research data repositories in 2015: A re3data analysis. *D-Lib Magazine 23*, 3/4 (2017).

[17] Koesten, L. M., Kacprzak, E., Tennison, J. F. A., and Simperl, E. The trials and tribulations of working with structured data: a study on information seeking behaviour. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, ACM, pp. 1277–1289.

[18] Madhavan, J., Ko, D., Kot, Ł., Ganapathy, V., Rasmussen, A., and Halevy, A. Google's deep web crawl. *Proceedings of the VLDB Endowment 1*, 2 (2008), 1241–1252.

[19] Nature scientific data. https://www.nature.com/sdata, 2018.

[20] Neumaier, S., Umbrich, J., and Polleres, A. Lifting data portals to the web of data. In *WWW2017 Workshop on Linked Data on the Web (LDOW2017)* (Perth, Australia, 2017).

[21] Noy, N., and Brickley, D. Facilitating the discovery of public datasets. https://ai.googleblog.com/2017/01/facilitating-discovery-of-public.html, 2017.

[22] Ohno-Machado, L., Sansone, S.-A., Alter, G., Fore, I., Grethe, J., Xu, H., Gonzalez-Beltran, A., Rocca-Serra, P., Soysal, E., Zong, N., and Kim, H.-e. Datamed: Finding useful data across multiple biomedical data repositories. *bioRxiv* (2016).

[23] Open data network. https://www.opendatanetwork.com/.

[24] Perego, A., Friis-Christensen, A., Vaccari, L., and Tsinaraki, C. DCAT-AP to schema.org mapping. Unofficial draft, 2018.

[25] Pujara, J., Miao, H., Getoor, L., and Cohen, W. Knowledge graph identification. In *International Semantic Web Conference* (2013), Springer, pp. 542–557.

[26] Quandl. https://www.quandl.com.

[27] Rastogi, V., Machanavajjhala, A., Chitnis, L., and Sarma, A. D. Finding connected components in map-reduce in logarithmic rounds. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on* (2013), IEEE, pp. 50–61.

[28] RDF 1.1 Concepts and Abstract Syntax. https://www.w3.org/TR/rdf11-concepts/.

[29] Rueda, L., Fenner, M., and Cruse, P. Datacite: Lessons learned on persistent identifiers for research data. *IJDC 11*, 2 (2016), 39–47.

[30] Sansone, S.-A., Gonzalez-Beltran, A., Rocca-Serra, P., Alter, G., Grethe, J. S., Xu, H., Fore, I. M., Lyle, J., Gururaj, A. E., Chen, X., et al. DATS, the data tag suite to enable discoverability of datasets. *Scientific data 4* (2017), 170059.

[31] Suchanek, F. M., Sozio, M., and Weikum, G. Sofie: a self-organizing framework for information extraction. In *Proceedings of the 18th international conference on World wide web* (2009), ACM, pp. 631–640.

[32] Varda, K. Protocol buffers: Google's data interchange format. Tech. rep., Google, 6 2008.

[33] Wang, J., Aryani, A., Wyborn, L., and Evans, B. Providing research graph data in JSON-LD Using Schema.org. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17 Companion, International World Wide Web Conferences Steering Committee, pp. 1213–1218.

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