Dataset or Not? A study on the veracity of semantic markup for dataset pages

Tarfah Alrashed[1], Dimitris Paparas[2], Omar Benjelloun[2], Ying Sheng[2], and Natasha Noy[2]

[1] CSAIL, MIT, USA
tarfah@mit.edu
[2] Google Research, Google, USA
{dpaparas, benjello, yingsheng, noy}@google.com

Abstract. Semantic markup, such as `Schema.org`, allows providers on the Web to describe content using a shared controlled vocabulary. This markup is invaluable in enabling a broad range of applications, from vertical search engines, to rich snippets in search results, to actions on emails, to many others. In this paper, we focus on semantic markup for datasets, specifically in the context of developing a vertical search engine for datasets on the Web, Google's Dataset Search. Dataset Search relies on `Schema.org` to identify pages that describe datasets. While `Schema.org` was the core enabling technology for this vertical search, we also discovered that we need to address the following problem: pages from 61% of internet hosts that provide `Schema.org/Dataset` markup do not actually describe datasets. We analyze the veracity of dataset markup for Dataset Search's Web-scale corpus and categorize pages where this markup is not reliable. We then propose a way to drastically increase the quality of the dataset metadata corpus by developing a deep neural-network classifier that identifies whether or not a page with `Schema.org/Dataset` markup is a dataset page. Our classifier achieves 96.7% recall at the 95% precision point. This level of precision enables Dataset Search to circumvent the noise in semantic markup and to use the metadata to provide high quality results to users.

Keywords: datasets, dataset search, semantic markup

1 Introduction

As the Web has grown in size and complexity, finding specialized information has become more challenging. Generalist search engines such as Google and Bing do well on common queries, but start to reach their limits when users are looking for content of a specific type or in a specific domain [3]. Vertical search engines have filled that niche by targeting domain-specific content and enabling users to explore it in a more structured way [24]. To illustrate, contrast the standard "10 blue links" Web search results with the experience offered by Pinterest for images, by Amazon for products, or by Google Scholar for publications. Generalist search engines are catching up by offering vertical-specific experiences for certain domains by showing, for example, custom results for jobs, recipes, or events.

Datasets are an important category of such specialized content [4]. With the adoption of open data in governments at all levels and with the scientific community encouraging data publication best practices [8], there is now a treasure trove of public datasets to help us understand the world, advance science, inform decision making, and enable social change. However, as the number of datasets continues to grow, it has become increasingly difficult to find relevant datasets using traditional search engines [15]. There are now thousands of dataset repositories on the Web with tens of millions of datasets [2]. Our team built Dataset Search [20], a tool that provides a single entry point for users to find datasets across all these repositories. Dataset Search is an interface over metadata of datasets from these different repositories and individual pages describing datasets. It relies on semantic markup in `Schema.org` and DCAT [3] to identify pages that describe datasets and their salient features of a given dataset, such as its name, description, license, and spatial and temporal coverage.[4]

`Schema.org` has become prevalent on the Web as a way to express the semantics of Web page content: it is present on more than 30% of Web pages [10]. It has enabled a wide range of applications and is indispensable for many search engines. However, in building Dataset Search, we discovered that we cannot always take `Schema.org/Dataset` markup at face value: pages may include this markup erroneously or for the purposes of search-engine optimization [2]. This problem is likely exacerbated by many, often vague, definitions of what constitutes a dataset. For example, Renear and colleagues analyzed the similarities and differences among dataset definitions in scientific literature [23].

In this paper, we analyze the scale of the problem, focusing in particular on whether pages with `Schema.org/Dataset` markup actually describe datasets. We then develop a method to mitigate the effect of misrepresented semantics by training a model to identify "true" dataset pages automatically. `Schema.org` is an important signal in interpreting Web pages, but we may need additional processing to ensure that markup that designates dataset pages is reliable.

The task of identifying dataset pages automatically presents unique challenges. First, datasets can cover any subject matter from core sciences to art, to real estate, to politics—or anything else. Therefore, for datasets the subject itself is not a distinguishing characteristic, and we cannot rely on domain-specific terminology either to include or to exclude a page. Second, the presence of terms such as "dataset" or "data" is not unique to pages that describe a dataset. For instance, there are thousands of tutorials and online courses for data scientists that discuss how to work with data and are not dataset pages. Third, there are no definitive structural cues for pages that describe datasets: a page with a table or a link to a CSV file may or may not represent a dataset.

[3] https://www.w3.org/TR/vocab-dcat-2/

[4] We use "semantic markup" to refer to the `Schema.org` metadata embedded in Web pages. The data representation may not technically be in a markup format, such as RDFa or Microdata, but could be embedded JSON-LD instead.

These difficulties are not unique to datasets. Other types of creative work, such as blogs and instructional materials, can cover any subject matter. The approaches that we discuss should be applicable to these verticals too.

Specifically, in this paper we make the following contributions:

- We motivate the need to address the veracity of semantic markup for datasets by analyzing cases where it does not correspond to the content of a page.
- We present a deep neural network that identifies whether a Web page with `Schema.org/Dataset` is a dataset page. To the best of our knowledge, this is the first model to focus on dataset pages.
- We demonstrate that our model outperforms the state-of-the-art classifiers for vertical and functional classification when those are used for dataset pages.
- We publish the following artifacts resulting from this research[5]:
  - A dataset of 223K URLs of pages with `Schema.org/Dataset` markup from 4.5K hosts, labeled as describing a dataset or not.
  - Source code with model configurations that can be used to retrain them.

2 Datasets and Dataset Pages: Problem definition

The key to our work is the definition of what constitutes a *dataset* and a *dataset page*. `Schema.org`, for example, defines a dataset as a "A body of structured information describing some topic(s) of interest." This definition is fairly general, and a likely culprit in the lack of clarity among metadata authors on what constitutes a dataset (cf. Section 4). The dataset definition in DCAT is similarly general.

We use the following definitions to clarify the scope in the context of datasets on the Web:

- A **dataset** is a collection of data items reflecting the results of such activities as measuring, reporting, collecting, analyzing, or observing.
- A **data item** itself can be an image, a number, a sentence, a structured object, another dataset. This list of types of data items is not exhaustive.
- A **dataset page** is a Web page that describes a dataset.

For example, a dataset can be a collection of values from a sensor, a set of labeled images, a set of sentences annotated with entities, or a set of survey responses. At the same time, an individual data item, such as a single measurement value, is not (usually) itself a dataset. Similarly, a collection of data items that can be derived computationally from first principles (i.e., a table of prime numbers or a table converting measurements) is not a dataset. A page that both describes and analyzes a dataset and provides links to a dataset download is a dataset page. A similar page where the dataset is embedded in the page as a table is also a dataset page. However, a page that has only a table, with no description of the table and no metadata, is not a dataset page.

Finally, we can informally think of a dataset page as any Web page that a user of a vertical dataset-search engine, such as Google's Dataset Search [20], might expect to see in the results.

[5] Available at www.doi.org/10.34740/kaggle/dsv/2407935

Note that this definition does not require the page to have any semantic markup identifying it as a dataset.

Classification problem. Let W be the set of all Web pages. Each Web page w is represented as a tuple $w = (u, c, m)$ where u is its URL, c its content, and m its (possibly empty) semantic markup. Let D be the set of all dataset pages. The *dataset-classification problem* is: given w, determine whether $w \in D$.

In this paper we restrict our attention to the set M of all Web pages with `Schema.org/Dataset` markup and we study the following problem (Figure 1): Given a page with markup, $w \in M$ (i.e., m is not empty), determine whether $w \in D$ (i.e., whether w is a dataset page).

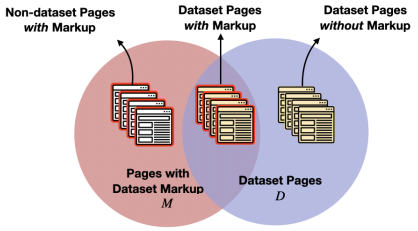

Fig. 1. Dataset pages and pages with `Schema.org/Dataset` markup on the Web. Not all pages with `Schema.org/Dataset` markup are dataset pages. Our goal is to identify the set w, where $w \in M \cap D$, as accurately as possible.

3 Related Work

Earlier analyses of the quality of linked open data and `Schema.org` highlighted common errors and ways to address them. For instance, Meusel and Paulheim proposed ways to fix errors automatically based on schema definitions [18]. The work on Pedantic Web discussed ways to use syntactic validation or reasoning to improve metadata [12]. The problem that we focus on is essentially that of a wrong assignment of semantic type, which these approaches do not address.

We can look to to other research areas for approaches to identify the type of content on the page: Web page classification and leveraging semantic markup in classification applications.

Web page classification. The key relevant approaches to classifying Web pages are topic classification, functional classification, and spam classification [22].

Topic classification categorizes Web pages based on their topic or subject (e.g., whether a page is about "news" or a "movie") for topic-specific search engines and Web content management [22]. Approaches to topic classification range from using only the URL [1, 11], to using the page content [5, 27], to including the structure as well as the content [14].

Functional classification determines the role that a Web page plays (e.g., a page can be a "course page" or a "faculty page"). Choudhury and colleagues used both textual content and hyperlinks to determine the role of a page [5]. Baykan and colleagues [1] devised a URL-based classifier for university pages, and argued that URL-based classification is preferable to content-based classification when Web pages need to be classified before they are fetched.

Datasets on the Web can cover any topic and be part of a page that plays any role (e.g., a dataset in a course page). Thus, neither topic nor functional classification applies directly for this domain; that is, a classifier that relies on

specific features of a vertical or a function might not work as well on identifying dataset pages. Others have also demonstrated that these types of classifiers might not work well for a different type of analysis [21].

A spam classifier [19] may help identify some of the non-dataset pages that claim to be datasets. But we cannot rely solely on a spam classifier: a page with dataset markup can be a valid (non-spam) page, but not a dataset page.

Semantic Markup in Web Classification. An alternative to classifying Web pages based on their URL, content, or structure is to use semantic markup like `Schema.org`. Krutil and colleagues used `Schema.org` annotations to classify pages into genres and micro-genres [16]. They argued that assigning Web pages to one or more predefined category labels would increase the precision of Web search. But their work makes the assumption that `Schema.org` annotations are used correctly, which is not always true in practice.

Semantic markup was also used to construct a database of events from the Web [25]. In this work, authors recognized that they cannot trust all event annotations as describing actual events. To address this problem, they manually identified large websites that use semantic markup incorrectly and removed them from their training set. Dataset pages have much more diverse structure than event pages. Thus, we can create similar training data by including pages that have semantic markup and excluding sites with known incorrect markup, but our features and model will be quite different than those for events.

4 The veracity problem for datasets

We performed a manual analysis of whether or not pages with `Schema.org/Dataset` markup were dataset pages, in the context of building and maintaining Dataset Search's corpus. To ensure the quality of search results, we regularly monitor sites that produce a large number of new pages with `Schema.org/Dataset` (our threshold was several hundred new such pages in a week) and verify whether or not these pages are dataset pages (cf. Section 2). We also enabled Dataset Search users to report pages that are not datasets. We verify such reports and, if warranted, exclude these pages or sites from the corpus. From this analysis, we collect a list of regular expressions that captures the URLs that are not dataset pages, a "denylist". As of today, this list captures hundreds of internet hosts.

Sometimes, the decision is straightforward (e.g., a page describing a gadget is not a dataset page); sometimes it is more of a judgement call driven by what we believed users expect to see in the search results (e.g., a real estate listing that has a table with tax history for a property).

The following are categories of pages with `Schema.org/Dataset` markup that were not dataset pages. This list is not exhaustive; rather, these categories are the types of pages that we have encountered multiple times in our corpus analysis:

Product pages: pages that describe a product or a collection of products, ranging from books to industrial supplies to real estate.

Data points: individual data points rather than datasets, such as stock price for a specific stock on a specific date; weather on a specific date in a location; lottery results for a specific date.

Information about items: pages describing a company or a medical practice.
Conversion tables: conversion between measurement systems or currencies.
Lists of terms: a collection of synonyms for a word.
Class exercises: pages with exercises and solutions for a class.
Explanations of an item: pages that provide a definition of an item or a tutorial on how to use it, such as a page describing the uses of a specific file extension (this particular example surprisingly prevalent).

The incorrect use of semantic markup in these cases is not necessarily malicious or intentionally misleading. "Dataset" is such a general term that providers may feel that it can apply to almost anything—making the task of creating a useful dataset search engine that much harder.

The process that we described in this Section clearly does not scale because it requires regular manual inspection of pages. It is also not comprehensive: we were able to examine only hosts with many new datasets; smaller sites could still appear in search results and make user experience worse. Thus, we built a classifier for dataset pages that enabled continuous comprehensive analysis and quality checks. It also provided a quantitative measure of the scale of the problem (see Section 6.4 for the results).

5 Dataset Classifier

We now discuss (1) the feature selection, (2) the creation of labeled data for training, testing, and evaluation, and (3) the details of the classifier. At a high level, we used a manual analysis to create a set of labeled data, experimented with a variety of structured and unstructured page features, and used an internal AutoML implementation to select and train a suitable model.

5.1 Feature selection

Dataset pages come in various shapes. Figure 2 (left) shows three examples of dataset pages. The first page consists of a description and a list of files, while the other two represent datasets using tables or charts. Some repositories like data.gov, have a consistent layout and structure for all their dataset pages. Other dataset pages, like the ones on GitHub, do not have a standard structure.

We trained our classifier on a combination of features extracted from both the HTML content and the semantic markup of Web pages.

URL and HTML Content Features

Early work on Web page classification [1, 11] considered only properties related to URLs. While URL-based classification does not provide ideal accuracy, this approach eliminates the need to analyze the page itself.

Repositories often add their dataset pages under a /dataset or a /catalog path and URLs may have terms indicating that the page is about a dataset (dataset, opendata, etc.). We capture the URL in tokenized form after pre-processing it to drop the domain. We exclude the domain because the number of distinct domains in our training set is orders of magnitude smaller than the number of pages, thus including it will overfit the model for specific domains.

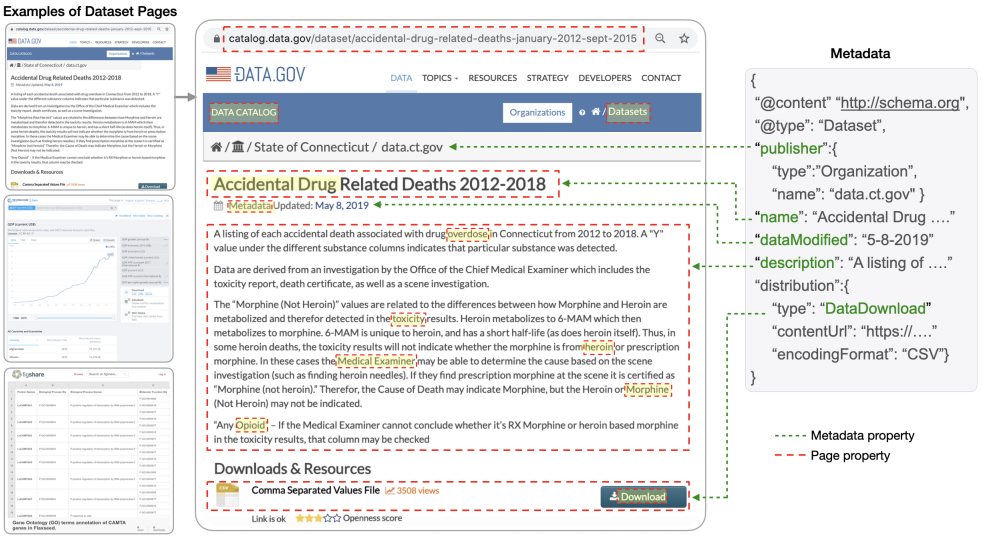

Fig. 2. An example of a dataset page, with highlighted page and metadata properties, and a few additional examples of dataset pages of varying structures.

To build a robust model that can handle a wide variety of dataset page structures, we need a representation of the page contents that does not depend on the page structure. We use a keyword extractor for Web pages to generate a vector of *prominent terms* and used them as features. Prominent terms are a collection of terms (unigrams/bigrams) with associated weights in the [0,1] scale. Each weight represents how relevant a term is for the page in question. To compute the weight of each term appearing in the document, we use a proprietary scoring model similar to the one proposed by Xiong and collueges [26]. Prominent terms consist of the top 100 terms resulting from that scoring.

Metadata Features

We used our observations from the manual analysis (Section 4) and the frequency of properties in the corpus [2] to select the `Schema.org` properties to use as features. Table 1 summarizes those properties. It also captures whether we used the values of a property (Section 5.4 discusses how we processed them) or simply the presence of the property in the metadata. For example, `name` and `description`[6] are required for every dataset and their content was the most useful source of information. Presence of download information through `distribution`, `encodingFormat` and `fileFormat` were likely to be a signal of high quality metadata, and thus, a valid dataset page. Similarly, the content of the provider or creator properties provided additional features.

We trained classifiers on two sets of features: a lightweight classifier (denoted DC-L) that used only `name` and `description` as metadata features and a full classifier (DC-F) that used all the features from Table 1. We found that both

[6] Unqualified property names such as `name` belong to the `Schema.org` namespace.

DC-F and DC-L achieved the same very high accuracy (Section 6), indicating that `name` and `description` together with prominent terms contain enough information to correctly classify a page.

Table 1. The features that we used for the two classifiers. For metadata features, the table captures whether we used the value of the property or just its presence (bool).

	Property Value	$DC-L$	$DC-F$
Non metadata features			
URL (without domain)			✓
Prominent Terms		✓	✓
`Schema.org` features			
`name`	value	✓	✓
`description`	value	✓	✓
`distribution`	bool		✓
`encodingFormat, fileFormat`	bool		✓
`provider, publisher`	value		✓
`author, creator`	value		✓
`doi`	value		✓
`catalog`	bool		✓
`dateCreated`	bool		✓
`dateModified`	bool		✓
`datePublished`	bool		✓

5.2 Labeled Data

We created an annotated dataset by sampling Dataset Search's corpus `Schema.org/Dataset` markup, which included millions of pages from thousands of domains [2]. We then labeled the data as either a dataset page or not.

We made the following assumption when labeling the data: If a host had pages with `Schema.org/Dataset` then either *all* of those pages were dataset pages or *none* of them were. Therefore, it was sufficient to sample a few pages from a site with semantic markup to label all pages from that site as positive or negative examples. Our earlier analysis showed that this assumption was almost always true: If website editors added `Schema.org/Dataset` to their pages, they either uniformly added it correctly or uniformly misused it.

Labeling process. All samples from our "denylist" (Section 4) were labeled as non-datasets. Additions to the denylist take place regularly and two team members, a proposer and an approver, must agree with the addition; and thus transitively, with the "not-dataset" label too.

For the remaining samples, we followed an in-house data-labeling approach. We formed a team of 7 raters —some of the authors of the paper and other colleagues— that reviewed the definition from Section 2 and together analyzed a number of positive and negative examples in order to align our ratings. We then

grouped the samples by host and partitioned them in 7 batches. We assigned each batch to a rater who labeled all hosts in it as "dataset", "not-dataset", or "unclear". We then had a follow up session where we discussed all "unclear" hosts, eventually labeling them as "dataset" or "not-dataset".

Sampling process. We used three sources of labeled pages. First, we randomly sampled pages that we had previously added manually to our denylist. Second, we sampled pages from the top 50 hosts in terms of the number of pages with `Schema.org/Dataset`. Finally, we obtained a random sample of about 1000 hosts from our corpus. Our labeled set had 223K pages, split almost equally between positive and negative examples.

We split the data into training, validation, and test sets with a rough ratio of 70:15:15, respectively. We now discuss the approaches that we used to ensure that the labeled set as a whole and the distribution between these subsets was balanced and diverse.

Balanced host representation. Because some hosts have millions of dataset pages and some have only a handful, we balanced the number of pages from each host in the labeled sample to avoid overfitting for a specific host. If a host had fewer than 100 pages with `Schema.org/Dataset`, we included all of them. For hosts with 101 to 2500 pages, we used a sampling rate of $10/\sqrt{\text{pages in host}}$, and for hosts with more than 2500 such pages we used a sampling rate of 500/pages in host. Thus, the number of samples from a host is increasing with the number of pages in the host and capped at 500.

Balance in test vs training set. Pages from the same data repository usually look similar because they are generated by the same code from a data catalog or a database. Thus, if we train a model on a subset of pages from a repository R and then use pages from R in our test data, it will be too easy for the model to identify them. To prevent the effects of memorization [14], we ensured that pages from the same host were either all in the training or all in the test set.

Pages in different languages. Dataset Search's corpus has pages in almost 100 languages. However, it was not practical for us to create a labeled set with balanced number of positive and negative examples in all languages. Indeed, our initial random sample contained a small number of pages in a specific language only as negative examples, causing the model to learn that all pages in that language are not datasets. In our final labeled set, we included pages in the five most frequently used languages (English, Chinese, Spanish, German, French). For other languages, we used the Google Translate API to translate the content of the relevant features (Table 1) into English and treated them as English pages.

5.3 Classifier Details

We trained multiple classifiers, starting with a rich set including all features in Table 1 (DC-F) and then experimenting with gradually smaller feature sets. The end result of this process was a lightweight classifier trained on only three features: `name`, `description`, and prominent terms (DC-L).

To pick an optimal model and to tune hyperparameters, we used a version of AdaNet [6], a framework to analyze and learn neural networks (NNs), to search over the space of Ensemble Estimators combining DNNs and Linear Estimators. We obtained the following models:

DC-L model. An ensemble estimator combining:

1. A feed-forward NN with one fully connected layer of 186 hidden units, implemented using TensorFlow DNNEstimator.[7] The architecture uses a Scaled Exponential Linear Unit (SeLU) activation function,[8] a dropout rate of 0.28673, and has Batch Normalization enabled.
2. A Linear estimator implemented using TensorFlow LinearEstimator.[9]

For ensembling we used AutoEnsemble[10] and optimized for sigmoid cross entropy loss[11] using the Adam Optimizer [13] with learning rate=0.00677, $\beta_1 = .9$, $\beta_2 = .999$, and gradients clipped using a clip norm of 0.00037. We used a batch size of 128 and trained to convergence (45k steps) using a custom trainer built on TensorFlow Extended.[12]

DC-F model. An ensemble estimator combining:

1. A feed-forward NN with three fully connected layers of 329, 351, and 292 hidden units respectively. Dropout rate is 0.08277 and Batch Normalization is disabled. The rest of the details are the same as for DC-L.
2. A Linear estimator implemented using TensorFlow LinearEstimator.

For ensembling we followed the same approach as for DC-L but with a learning rate of 0.00076 and a clip norm of 0.25035.

5.4 Feature processing

To process the values for text features in Table 1, we tokenized on white-space and punctuation and then combined words into unigrams and bigrams, presented to the model as a bag of words. This approach considers occurrences of words in isolation, as well as pairs of words. We selected tokens to retain in a vocabulary based on adjusted mutual information between each token and the label. This filtering reduced the vocabulary size to 3-4% of the original. We used two out-of-vocabulary hash buckets for any tokens that are not in the vocabulary. For the DNN sub-graph, the tokens were embedded using a learned embedding of size proportional to the log of the vocabulary size. To create a fixed dimensional input vector, the variable length bags of tokens were combined using the weighted sum of the embedding weights, divided by the square root of the sum of the squares of the weights [13].

[7] https://www.tensorflow.org/api_docs/python/tf/estimator/DNNEstimator
[8] https://www.tensorflow.org/api_docs/python/tf/keras/activations/selu
[9] https://www.tensorflow.org/api_docs/python/tf/estimator/LinearEstimator
[10] https://adanet.readthedocs.io/en/v0.9.0/_modules/adanet/autoensemble/estimator.html
[11] https://www.tensorflow.org/api_docs/python/tf/nn/sigmoid_cross_entropy_with_logits
[12] https://www.tensorflow.org/tfx
[13] https://www.tensorflow.org/api_docs/python/tf/nn/embedding_lookup_sparse

6 Dataset classifier: Evaluation and Results

Our experiments measure the accuracy of the dataset classifiers, DC-F and DC-L. We also compare them to three state-of-the-art Web page classifiers, which we implemented as our baseline methods.

6.1 Baseline Methods

Section 3 presented a number of approaches to Web page classification. Vertical and functional classifications are two of the main types that researchers have studied extensively [22]. We applied three state-of-the-art methods to classify dataset pages: One functional classifier and two vertical classifiers (content-based and content and structure based). We trained all three models on the labeled data described in Section 5.2. We evaluated our dataset classifier against classifiers developed for different purposes because, to the best of our knowledge, no classifiers have been developed specifically for datasets.

Functional Classifiers (content-based) Choudhury et al. [5] evaluated numerous methods for categorizing Web pages based on their role by using the content and the hyperlink text on the page. They used the WebKB dataset [7] to train and test a number of models, mainly Multinomial Naive Bayes (NB) and Support Vector Machine (SVM) to classify Web pages from computer science departments of several universities into multiple categories: project, course, faculty, department. We implemented their approach by training the SVM and NB models on the content and hyperlink text. We chose to apply this method because SVMs have worked well for text classification, due to the large dimensionality of the feature space, and the sparsity of feature vectors [5]. In addition, SVM and NB have been used as baselines for many content-based Web classification [22].

Vertical Classifier (content-based) The second baseline method that we use is inspired by the work from Zhao et al. [27]. They proposed a network-classification model based on deep learning that takes the title and description ("short text"), and the textual content of the Web page ("long text") as features. They classified pages from Web portals, like Sina and NetEase, into a number of vertical categories (entertainment, art, etc). Instead of considering the whole content text as their only input feature, the authors argued that considering the title and description of the page yielded a better classification accuracy [9]. Following their approach, we trained three variants of the model: Using the title and description (short text), using only the content text (long text), and using the title, description and the content text (short/long text).

Vertical Classifier (content- and structure-based) As an alternative to classification based only on content, we also considered RiSER [14], a model that incorporates both the structure and content of pages to classify machine-generated emails into verticals (hotel, bill, etc). Such emails are typically short, and have rich HTML structure. Instead of considering the whole document content as a feature, RiSER extracts the first 200 textual terms from the DOM-tree, with the XPaths that lead to them. RiSER authors trained their model on a large

corpus of anonymized emails received by users of Gmail and evaluated it on two different classification tasks. To retrain RiSER, we used the html extracted from the data presented in Section 5.2. However, unlike emails, the assumption that the first 200 words will contain a useful signal is not true for webpages. To address this, we pre-processed all html and removed text with tags that are frequently irrelevant, such as buttons (e.g., "login") and menus (e.g., "home"). We then trained and tested Riser using the results of this cleanup.

6.2 Metrics

We compare the two variants of our dataset classifier, namely DC-F and DC-L, with the three baseline methods using the area under the precision-recall curve (AUC-PR). Classifiers are often evaluated using area under the receiver operating characteristic curve (AUC-ROC). However, ROC curves may provide an excessively optimistic view of the performance for highly skewed domains [14]. The AUC-PR is similar to AUC-ROC in that it summarizes the curve with a range of threshold values as a single score. The score can then be a point of comparison between different models on a binary classification problem where a score of 1.0 represents a model with perfect skill. AUC-PR alone is insufficient to evaluate our models because applications like dataset search require very high precision. Indeed, returning non-dataset pages to users searching for datasets would be a poor user experience. Therefore, we evaluate our models based on their recall at a fixed level of high precision, specifically at a precision equal to 95% (@$P95$). In addition to the AUC-PR and recall @$P95$ metrics, we report the F1-score @$P95$ (in both Table 2 and Figure 3) for the dataset classifier variants and baseline models. F1-score is the weighted average of precision and recall. Thus, comparing the models based on their F1-score should be the same as comparing them using the recall @$P95$ metric.

We trained multiple classifiers using different combinations of features and from this process we concluded that name, description, and prominent terms are the most important ones for the classification.

6.3 Results

Table 2 shows the results for the three baseline methods. For the functional classifiers, SVM performed slightly better than the Multinomial NB, with an AUC-PR of 77, but a low recall @$P95$. Classifiers tailored to identify the role of a page are not suitable to identify pages that contain datasets.

Vertical classifiers performed better than the functional ones. RiSER achieved a higher AUC-PR than all other baselines, however, it was still only able to correctly identify 66% of dataset pages at @$P95$.

Figure 3 shows the results for DC-F and DC-L, compared to the best model for each baseline method from Table 2: Content and link based SVM, content based (long text) DNN, and RiSER. DC-L outperforms the functional SVM classifier by an additive factor of 73%, the vertical DNN classifier by 41%, and RiSER by 31% on recall @$P95$. The results show that using both metadata

Table 2. The performance of DC-L, DC-F, and baseline classifiers.

Type of Classifier	Model	AUC-PR	R@P95	F1@P95
Functional (Content based)	Multinomial NB	0.67	0.22	0.35
	SVM	**0.77**	**0.24**	**0.38**
Vertical (Content based)	DNN (Short)	0.85	0.49	0.63
	DNN (Long)	**0.90**	**0.56**	**0.69**
	DNN (Short/Long)	0.88	0.55	0.68
Vertical (Content & Structure based)	**RiSER**	**0.96**	**0.66**	**0.77**
Dataset (Content & Metadata based)	**DC-L**	**0.99**	**0.97**	**0.96**
	DC-F	**0.99**	**0.96**	**0.96**

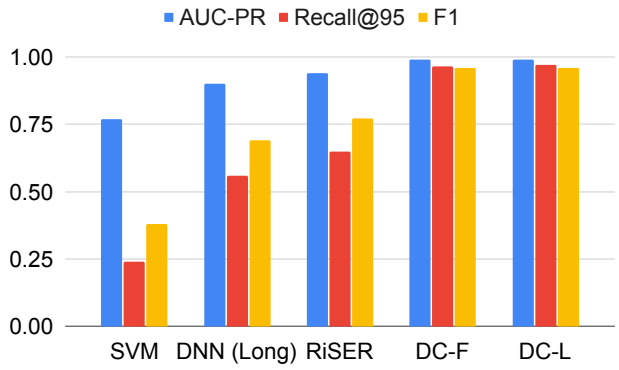

Fig. 3. Performance (AUC-PR, Recall and F1 score at Precision=95) comparisons between our dataset classifier variants (DC-F and DC-L) and the baseline methods.

properties and page properties as features provides performance gains over using only page properties.

The AUC-PR metric is also improved for the DC-F and DC-L variants compared to the baseline models. DC-L outperforms the SVM classifier by 22%, the vertical DNN classifier by 9%, and the RiSER model by 3%.

We manually inspected the 1% of web pages that the classifier predicted incorrectly. We found that some of these pages are information pages that describe a company, a product, a place, or a person, others are online web page translators, climate web pages, and pages about scientific papers or stock data. Some of these pages are tricky to label, such as the stock data and the climate pages. Others are obvious non-dataset pages, like the info pages.

6.4 Corpus-Level Analysis

We used DC-L to evaluate the veracity of dataset markup in our entire corpus of more than 600M pages on the Web with the minimal `Schema.org/Dataset` markup (at least `name` and `description`). For pages in a language other than the five that we trained the classifier on, we first translated their features to English. We used the @P95 prediction threshold t for which our experiments achieved the highest recall and precision (Figure 3). We aggregated the results at the host level and classified hosts into those with more than half of their pages scoring above t (dataset hosts) and the rest (non-dataset hosts). We then assigned each

page the same label as its host, regardless of its individual classification score. This majority-vote smearing of the score ensured robustness against outliers.

To validate our majority-vote approach, we performed an additional evaluation of applying DC-L to the entire corpus. We randomly sampled 250 hosts classified by our majority-vote method as dataset hosts and another 250 hosts classified as non-dataset hosts, excluding hosts that were already in the labeled data. We again followed an in-house labeling approach (Section 5.2). We labeled these 500 hosts and determined recall and precision to be 99% and 96% respectively. The recall at host level is slightly higher than for individual pages, likely because the majority-vote aggregation accounts for outliers.

Figure 4 captures the distribution of dataset and non-dataset pages and hosts for the entire corpus. These numbers exclude the 10 largest hosts with `Schema.org/Dataset` on non-dataset pages as those hosts account for 40% of non-dataset pages and are true outliers. For comparison, the next 10 largest hosts by number of non-dataset pages account for only 7% of these pages. The results show that the majority of hosts (61%) and pages (84%) are non-dataset pages.

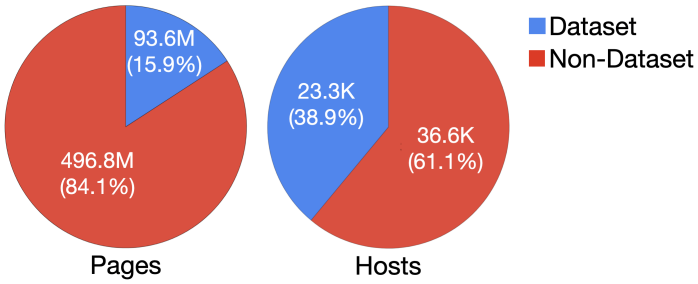

Fig. 4. Dataset vs non-dataset pages and hosts in the Web corpus with `Schema.org/Dataset` markup (excluding 10 largest non-dataset hosts).

7 Discussion and Future Work

In this paper, we analyzed the basic premise of whether semantic markup can be taken at face value. For datasets at least, this premise does not always hold true (Section 4). The difficulty of defining what a dataset is likely contributes to the problem. Having `Schema.org` define a dataset more precisely (similar to what we proposed in Section 2) may significantly alleviate the problem. However, it may be useful to evaluate more deeply how prevalent this problem is in other verticals, such as recipes, jobs, events, and so on.

7.1 Understanding Classification Results

Our classifiers outperformed the baseline classifiers (Section 6) because the dataset classification task cannot be reduced to one of the problem classes for which existing classifiers were designed. The functional classifiers, SVM and the Multinomial NB, aimed to identify the role of pages. The vertical classifier DNN was designed to identify the vertical or the topic of pages. Finally, RiSER focused

on classifying highly structured, short business-to-consumer emails. Dataset pages can serve multiple functions, may belong to many verticals, and are not uniform in size and structure.

Our results show that using only `name` and `description` as metadata features achieves essentially the same accuracy as using other properties or URL-based features. That is, knowing that the page is marked up with `Schema.org/Dataset` and seeing the value for name and description properties is often sufficient to determine whether it is a dataset page. Other properties, however, can provide important information about the quality of the page. Presence of download information through the `distribution`, `encodingFormat` and `fileFormat` properties may be a signal of high quality metadata. We also noticed that dataset pages often contain more complete metadata: they have 5.2 properties on average, whereas non-dataset pages have 2.57.

7.2 Quality versus Coverage

Our work focused on the problem of *quality* in the context of dataset search: identifying whether or not a page that has `Schema.org/Dataset` is a dataset page. For this problem, high recall and precision are essential. A dataset page with `Schema.org/Dataset` markup comes from a content provider who put in the effort to describe semantics of the page. Mis-classifying such a page as "not a dataset" will be both unfair to the content provider and detrimental to the quality of search results. Thus, we need high recall. At the same time, a page with dataset metadata that is not a dataset page would not be useful to a user searching for datasets. If there was a small number of such pages, we could simply ignore the problem because these pages would rarely show up in search results. However, it turns out that there are tens of millions of pages that contain semantic markup without actually describing a dataset. Thus, high precision is crucial to achieve high quality of search results: We want to ensure that non-dataset pages are unlikely to show up among results that contain predominantly true dataset pages. With AUC-PR of 99 (Table 2), the model we proposed is sufficiently accurate to decide automatically what pages to include in a dataset search engine.

There is a complementary problem to the problem of quality, that of *coverage* (Figure 1): given a Web page without `Schema.org/Dataset` metadata, decide whether this page is a dataset page. We did not address the problem of coverage in this paper, however, future work may address how we can use URL and content features to classify pages in this domain. Even if such a classifier does not have high recall, it can be useful to identify key data repositories that do not have `Schema.org` markup, and inform an outreach strategy towards influencers and repositories, as described by Noy and colleagues [20].

In this work, we focused on the veracity of the type of entity a Web page is about (i.e., `Schema.org/Dataset`), not on specific property values [17]. Just like type information, property values also cannot be taken at face value. It would be useful to explore the space where the property values are set incorrectly, intentionally or not, in order to continue improving the quality of applications that rely on semantic markup.

8 Conclusions

The goal of semantic markup such as `Schema.org` is to provide a machine-readable interpretation of the contents of a Web page. This markup has enabled a myriad of applications. However, as these applications begin to cover domains with ambiguous and more general artifacts, such as datasets, the markup becomes more noisy and less reliable. Specifically, we showed that providers often misunderstand when a page should be typed as a dataset page, which is an example of such general category. We propose a way to remedy the problem automatically by using semantic markup as an important signal but also considering other features to support it. Our work enables high quality results based on `Schema.org`, making use of the rich semantics, even in a domain where the markup is unreliable.

9 Data and code availability

We have made both the dataset discussed in Section 5.2 and code to train the models from Table 1 available through www.doi.org/10.34740/kaggle/dsv/2407935. However, we had to remove the *prominent terms* column from the dataset because the scoring model we used to obtain it is not publicly available. Interested users can replicate this scoring using the model by Xiong and colleagues [26].

Acknowledgments

We are grateful to Amy Skerry-Ryan, Katrina Sostek, Marc Najork, and Shiyu Chen for discussions on this work.

References

1. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely url-based topic classification. In: 18th International Conference on World Wide Web. p. 1109–1110. WWW '09 (2009). https://doi.org/10.1145/1526709.1526880
2. Benjelloun, O., Chen, S., Noy, N.: Google dataset search by the numbers. International Semantic Web Conference (2020)
3. Bozzon, A., Brambilla, M., Ceri, S., Fraternali, P.: Liquid query: Multi-domain exploratory search on the web. In: 19th International Conference on World Wide Web. p. 161–170. WWW '10 (2010). https://doi.org/10.1145/1772690.1772708
4. Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L.D., Kacprzak, E., Groth, P.: Dataset search: a survey. The VLDB Journal **29**(1), 251–272 (2020)
5. Choudhury, S., Batra, T., Hughes, C.: Content-based and link-based methods for categorical webpage classification (2016)
6. Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., Yang, S.: Adanet: Adaptive structural learning of artificial neural networks. In: International conference on machine learning. pp. 874–883 (2017)
7. Craven, M., McCallum, A., PiPasquo, D., Mitchell, T., Freitag, D.: Learning to extract symbolic knowledge from the world wide web. Tech. rep., Carnegie-mellon univ pittsburgh pa school of computer Science (1998)
8. Fenner, M., Crosas, M., et al.: A data citation roadmap for scholarly data repositories. Scientific Data **6**(1), 1–9 (2019). https://doi.org/10.1038/s41597-019-0031-8
9. Golub, K., Ardö, A.: Importance of HTML structural elements and metadata in automated subject classification. In: International Conference on Theory and Practice of Digital Libraries. pp. 368–378. Springer (2005)

10. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: evolution of structured data on the web. Communications of the ACM **59**(2), 44–51 (2016)
11. Hernández, I., Rivero, C.R., Ruiz, D., Corchuelo, R.: A statistical approach to url-based web page clustering. In: 21st International Conference on World Wide Web. p. 525–526. WWW '12 Companion (2012). https://doi.org/10.1145/2187980.2188109
12. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic web. LDOW **628**, 26 (2010)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kocayusufoglu, F., Sheng, Y., Vo, N., Wendt, J., Zhao, Q., Tata, S., Najork, M.: Riser: Learning better representations for richly structured emails. In: The Web Conference. p. 886–895. WWW'19 (2019). https://doi.org/10.1145/3308558.3313720
15. Koesten, L.M., Kacprzak, E., Tennison, J.F.A., Simperl, E.: The trials and tribulations of working with structured data: -a study on information seeking behaviour. CHI '17 (2017). https://doi.org/10.1145/3025453.3025838
16. Krutil, J., Kudělka, M., Snášel, V.: Web page classification based on schema.org collection. In: 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN). pp. 356–360 (2012)
17. Lin, B.Y., Sheng, Y., Vo, N., Tata, S.: FreeDOM: A transferable neural architecture for structured information extraction on web documents. In: ACM KDD. p. 1092–1102 (2020). https://doi.org/10.1145/3394486.3403153
18. Meusel, R., Paulheim, H.: Heuristics for fixing common errors in deployed schema.org microdata. In: The Semantic Web. Latest Advances and New Domains. Springer (2015)
19. Najork, M.: Web spam detection encyclopedia of database systems (2009)
20. Noy, N., Brickley, D., Burgess, M.: Google Dataset Search: Building a search engine for datasets in an open web ecosystem. In: The Web Conference. WWW'19 (2019). https://doi.org/10.1145/3308558.3313685
21. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: AEmpirical Methods in Natural Language Processing, EMNLP. p. 79–86. USA (2002). https://doi.org/10.3115/1118693.1118704
22. Qi, X., Davison, B.D.: Web page classification: Features and algorithms. ACM Comput. Surv. **41**(2) (Feb 2009). https://doi.org/10.1145/1459352.1459357
23. Renear, A.H., Sacchi, S., Wickett, K.M.: Definitions of dataset in the scientific and technical literature. American Society for Information Science and Technology **47**(1), 1–4 (2010). https://doi.org/10.1002/meet.14504701240
24. Shettar, R., Bhuptani, R.: A vertical search engine–based on domain classifier. International Journal of Computer Science and Security **2**(4), 18–27 (2007)
25. Wang, Q., Kanagal, B., Garg, V., Sivakumar, D.: Constructing a comprehensive events database from the web. In: 28th ACM CIKM (2019). https://doi.org/10.1145/3357384.3357986
26. Xiong, C., Liu, Z., Callan, J., Liu, T.Y.: Towards better text understanding and retrieval through kernel entity salience modeling. In: 41st ACM SIGIR (2018)
27. Zhao, Q., Yang, W., Hua, R.: Design and research of composite web page classification network based on deep learning. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). pp. 1531–1535. IEEE (2019)