A Dataset Search Engine for the Research Document Corpus

Meiyu Lu [#], Srinivas Bangalore [*], Graham Cormode [*], Marios Hadjieleftheriou [*], Divesh Srivastava [*]

[#]*National University of Singapore*
lumeiyu@comp.nus.edu.sg

[*]*AT&T Labs-Research*
{srini,graham,marioh,divesh}@research.att.com

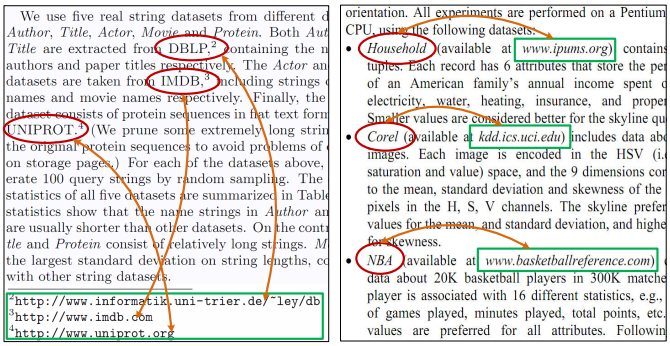
***Abstract*—A key step in validating a proposed idea or system is to evaluate over a suitable dataset. However, to this date there have been no useful tools for researchers to understand which datasets have been used for what purpose, or in what prior work. Instead, they have to manually browse through papers to find the suitable datasets and their corresponding URLs, which is laborious and inefficient. To better aid the dataset discovery process, and provide a better understanding of how and where datasets have been used, we propose a framework to effectively identify datasets within the scientific corpus. The key technical challenges are identification of datasets, and discovery of the association between a dataset and the URLs where they can be accessed. Based on this, we have built a user friendly web-based search interface for users to conveniently explore the dataset-paper relationships, and find relevant datasets and their properties.**

I. Introduction

Progress in science is made by proposing hypotheses, and evaluating these against appropriate datasets. The choice of a suitable dataset to validate on is key to the scientific method. Yet the selection process is typically haphazard: researchers recycle datasets they are familiar with, or have heard about through word-of-mouth. The reason is that we do not have good tools for *dataset discovery*: there is no widely used resource that indexes datasets, and the relationships between datasets and scientific research.

This stands in contrast to the (scientific) literature discovery problem: there are many tools and systems which help to find relevant related work, such as DBLP, PubMed, arXiv, Citeseer, Google Scholar, ACM Digital Library and IEEE CS Digital library. These present a search-engine like interface which allows users to search papers via keywords, or via faceted search on features such as author, venue etc. Results are presented in a structure-aware fashion, exposing attributes of the papers (authors, abstract, venue, date etc.) and enabling users to explore the citation network: which papers cited a particular paper, and which do they cite?

In this work, our goal is to build a comparable search engine for *datasets*. The uses of such a tool are manifold: it can enable a researcher to identify what are the key datasets and *de facto* benchmarks for a particular area (e.g. frequent pattern mining, or data anonymization) or domains (e.g. web search engine logs, social networks). For a target dataset, we can find what other datasets have also been used in the same paper, or



(a) Example 1 (b) Example 2

Fig. 1. Dataset Description Examples

what are the relevant datasets in the literature (i.e. being used together in many research papers). To support this aim, we have built a search engine for datasets, to aid dataset discovery and dataset usage analysis. We view such a tool as orthogonal to existing academic search engines, but equally important.

There are many challenges to overcome in meeting this goal. The first one is the identification of dataset entities from the underlying research documents. Although many information extraction approaches have been proposed and work successfully for some extraction tasks [5], there has been little prior effort to collate information on datasets. In contrast to person and organization name extraction from newspapers, which have certain features and usage patterns, in research literature the styles used to introduce datasets and provide URLs to them vary widely across authors and documents. Figure 1 shows two example styles in two different documents by the same author. In Figure 1(a), datasets are introduced within a paragraph, and associated URLs are listed as footnotes; in Figure 1(b), datasets are given in a list, with a corresponding URL listed parenthetically. There are many other styles, including description in flat text, specification within a table, and even references within a bibliography; this makes it difficult to apply rule-based information extraction for our case.

The second challenge is dataset resolution. References to the same dataset may vary widely across documents. For example, the DBLP data could be referred to as "DBLP Bibliography", "DBLP Dataset", and so on. Effectively recognizing the co-references is crucial for our dataset search engine. This re-

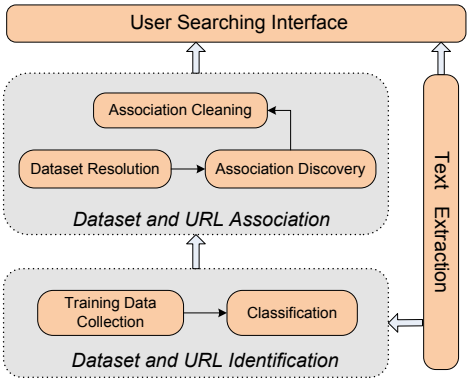


Fig. 2. System Framework

quires a good similarity measure which is able to give more weight to dataset-relevant words, while less to general words, so that "DBLP Bibliography" and "DBLP Dataset" can be recognized as the same dataset, while "DBLP Dataset" and "IMDB Dataset" are different.

The third challenge is discovery of the associations between datasets and URLs, i.e. which URL corresponds to which dataset. Usually, a URL for the dataset is listed nearby in the text, or in a footnote or bibliography entry. Extracting URLs from such references proves to be difficult, since the PDF (Portable Document Format) format renders these just as part of the text. We found pattern based approaches to extract associations [2] were not helpful for this problem.

In this demonstration, we propose a framework to effectively identify datasets and the corresponding URLs within the scientific research corpus, and provide a user friendly interface for searching datasets and exploring the relationships between datasets and the research documents.

II. System Overview

Figure 2 shows the framework of our system. It consists of three main components. In the *Dataset and URL Identification* stage, a set of training data are first collected, and a classifier is built over the collected data to recognize references to datasets, in the form of dataset names and URLs. In the example of Figure 1, identified dataset names are marked with a red ellipse, and dataset URLs with a green rectangle. The next step of *Dataset and URL Association* is to identify the links between dataset names and URLs, shown via the orange arcs in Figure 1. Finally, a *User Interface* in the form of a dataset search engine allows users to search for datasets and papers by keywords and explore the dataset-paper graph.

Below, we describe the first two phases of our system, i.e. the dataset and URL identification and association stages. The search engine functionalities will be introduced in Section III.

Text Extraction. The scientific community has converged on the Portable Document Format (PDF) as the *de facto* standard for sharing research papers. We make use of the PDFBox [1] tool to extract the text content from PDF documents. We collected a large corpus of documents by crawling the ACM Digital Library. We observed that most short papers do not include references to datasets, so we focused our attention

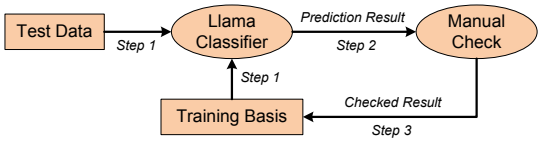


Fig. 3. Active Learning

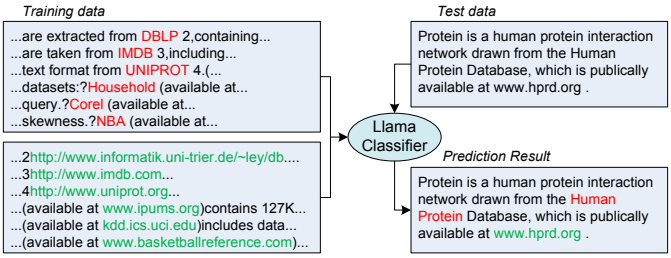


Fig. 4. Classification

on papers of 10 or more pages in length. As a preprocessing step, from each paper we extracted sections with experimental content by matching a regular expression (using words such as "Experimental", "Experiment(s)", "Empirical", "Performance Evaluation", "Results Analysis" in section headings), as datasets are primarily described in such sections. In total, we obtained a collection of experimental sections from 14,946 papers mainly from the Computer Science domain.

A. Dataset and URL Identification

We make use of the feature based, multiclass "Llama" classifier [3] to identify dataset names and references from research documents.

Training Data Collection. Given the wide variety of styles and keywords used to refer to datasets, we need a large training set to accurately train our classifier. However, the process of manually labeling thousands of documents, each containing thousands of words, is a very time-consuming one. Instead, we adopt *active learning* [4] to speed up the process, as illustrated in Figure 3. This iterative process lets us generate a high quality training set much more efficiently. In total, we built a training set of 1000 documents, containing 8,922 words labeled as datasets, and 2,280 accompanying URLs.

Classification. Our classifier uses a large number of features to decide whether a given word is a reference to a dataset. The main features are derived from the *context* around each word in question. Figure 4 shows a context of three words (counting punctuation) before and after each word. This generates a huge number of binary features: the word "from" immediately preceding the word in question is one feature; the word "available" two positions after the word is another. We also added additional features of the word itself that can help to indicate the name of a dataset, including (a) whether the whole word is in capital letters (b) whether the first letter is capitalized, and (c) whether the word is a URL (matching a hand-crafted regular expression for being a URL). These are based on the observation that initial capitals or fully capitalized words are commonly used to indicate dataset names.

Identification Effectiveness. Given the large number of datasets in use, it is important that the classifier should be able to identify novel datasets, even if these have not occurred within its training data. Figure 4 shows an example of this, where the dataset name *Human Protein* and accompanying URL `www.hprd.org` in the test data are correctly identified by the classifier, based on their contexts and features. We evaluated the precision and recall of this method at the phrase-level: if the test data contains a sequence of words labeled as identifying a dataset (e.g. "Internet Movie Database"), we score this as correct if any of these words are classified as dataset. Under this measure, our classifier achieves 82% precision and 59% recall for datasets, and 78% precision and 73% recall for relevant URLs, sufficiently high for our needs.

B. Dataset and URL Association

We now discuss our approach to the identification of associations between the dataset names and URLs recognized in the *Dataset and URL Identification* stage.

Dataset Resolution. Given a set of "dataset" words identified by the classifier, we define a *dataset phrase* as a sequence of consecutive dataset words. Dataset resolution is then performed over the identified dataset phrases. The first step is to normalize the phrases. Normalization entails removing stop words, punctuation, special characters, numerical prefixes and suffixes (often due to footnote numbers), then performing stemming, and finally making everything case insensitive. Once the data has been normalized we cluster phrases based on weighted Jaccard similarity by treating each phrase as a bag of words. To cluster phrases efficiently we create a graph where each phrase is a node, and an edge exists between two nodes if and only if the weighted Jaccard similarity between these nodes is larger than a given threshold θ. Finally, we compute the connected components of this graph. Each component defines a cluster and represents a dataset entity. We can make the graph sparser or denser by varying θ. In our experiments a Jaccard similarity of 0.6 gave very good results. Varying θ from 0.5 to 0.8 did not affect the clusters significantly.

An important step here is to assign weights to words, for the purpose of computing weighted Jaccard similarity. Ideally, as discussed in the challenges from Section I, we would like to give high weights to words associated with particular datasets, and low weights to more generic words. We notice that it is tempting to use phrase level IDF weights for this purpose (the phrase level IDF weight of a word is the inverse of the number of times this word appears in the collection of phrases), because in this scenario words that belong to very popular datasets tend to have small IDF weights. For example, words "DBLP" and "TREC" have IDF weight equal to 3.6, which is the same weight as the words "IMAGE" and "BENCHMARK". However, the word "ABOUT" and the word "TPCE" have IDF weight 12.6. This is a problem, because for popular datasets, the most important words lose their descriptive capacity. To account for this we observe that it is also important to consider the IDF of each word in the whole corpus of documents (as opposed to only within the corpus of dataset phrases that have been identified by the classifier). Intuitively, even if "DBLP" is a common word among datasets, it should not be as common as "BENCHMARK" or "ABOUT" across the whole document corpus. In the end, we use phrase level IDF weights for each word, and modify them by boosting the weights of words with high IDF weights across the document corpus and reducing them otherwise. We also boost the weight of a word for each occurrence of the word that has at least one capital letter (before normalization).

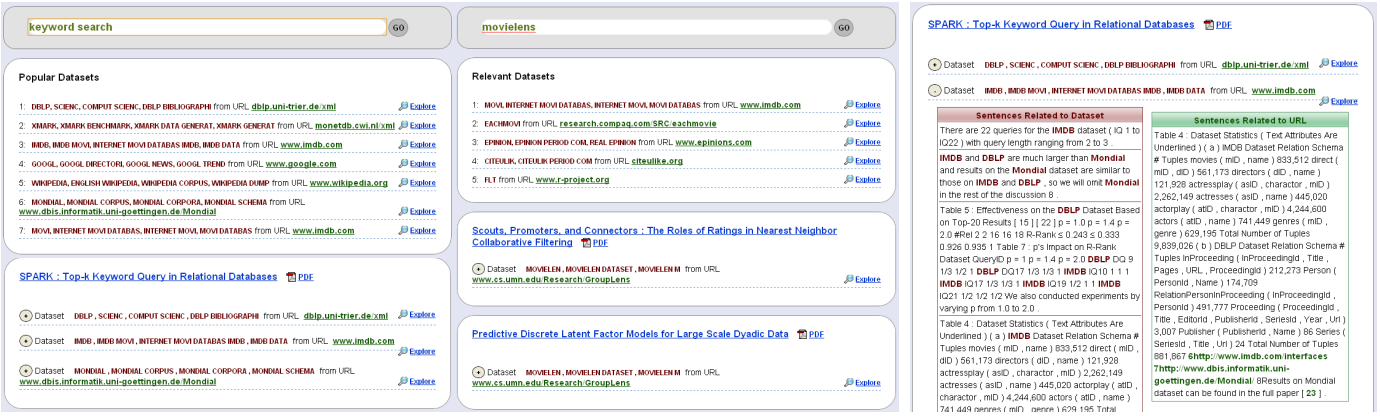
Association Discovery and Cleaning. After the dataset phrase clusters have been formed we associate URLs with each cluster. This is quite challenging, since URLs associated with a dataset may not occur in close proximity to the dataset name; this makes it difficult to apply pattern-based extraction of associations. Instead we make use of the observation that datasets and their URLs tend to co-occur in multiple documents. This motivates our scalable technique of using co-occurrence of datasets and URLs across multiple documents to discover associations. First, we tally the number of times a particular URL appears in the documents containing any of the phrases within a given cluster. We associate that cluster with the top-k URLs. The larger the tally of a given URL is, the higher is the confidence of the association. We also use normalization and edit distance to aggregate the tallies of similar URLs. In this way, each dataset cluster is associated with a set of URLs.

In the URL set discovered above for the dataset cluster, some top ranked URLs are actually not relevant to the associated dataset, such as `doi.acm.org` (prefix of the DOI link) and `www.computer.org/publications/dlib` (link for IEEE CS Digital Library), because these links appear in a large number of documents. To remove such noisy URLs, we tally the number of distinct dataset clusters that each URL is associated with, and then drop the URLs that are associated with too many dataset clusters; the intuition is that if a URL is associated with too many dataset clusters, its association with each cluster is not strong, even if it co-occurs frequently with this cluster.

After removing noisy URLs, the top-4 co-occurring URLs are maintained as representatives for each dataset cluster. This keeps the quality of the URLs high, since a dataset is not expected to be associated with too many distinct URLs. Similarly, we summarize each dataset cluster by choosing the top-k most frequent normalized phrases within the cluster. For example, the cluster containing phrases related to IMDB is summarized as "IMDB, IMDB MOVIE, INTERNET MOVIE DATABASE, IMDB DATA, IMDB DATASET". The representative URLs associated with this cluster are `www.imdb.com`, `www.imdb.com/interfaces` and `imdb.com`.

III. System Demonstration

Our demo (available at `http://db128gb-b.ddns.comp.nus.edu.sg/meiyu/datasetsearch/`) focuses on use cases from Section I. (1) Dataset discovery: identifying which datasets are popular and used by the documents in a given area; (2) Dataset usage: identifying documents where a



(a) Dataset Discovery (b) Dataset Usage (c) Detailed Interface

Fig. 5. System Demonstration Screenshots

given dataset has been used, and the other relevant datasets in those documents; and (3) Dataset exploration: exploring connections between documents and datasets.

Dataset Discovery. The first function we provide allows searching for datasets by domain concepts/terms, such as "keyword search" in Figure 5(a). Users can input keywords to retrieve a list of documents, or a paper title to retrieve exactly the document he/she wants to analyze. For ease of exploration, each returned document is shown with snippets indicating the dataset names it uses and their corresponding URLs. Moreover, a set of popular datasets in the queried domain are also returned to the users, by analyzing the common datasets among the returned documents. Figure 5(a) shows several popularly used datasets in "keyword search" domain, including *DBLP*, *XMARK*, *IMDB*, *WIKIPEDIA* and *MONDIAL*, as well as the specific datasets within the top-ranked document on this topic. Each dataset in the interface can be directly accessed by clicking on the URLs that have been extracted. To help the users better understand the datasets and dataset properties, for each returned document we also list the relevant sentences for each dataset and URL. As shown in Figure 5(c), the detailed sentences for *IMDB* can be expanded out.

Dataset Usage. The second function we provide is to explore by dataset. Given the dataset keywords entered by a user, our system returns a list of documents which have used this dataset. The user can then explore which papers (and hence, domains) have used the specific dataset. Figure 5(b) shows results for dataset "movielens". Although we only present the first two documents in the screenshot, it clearly shows that *movielens* dataset is used in collaborative filtering and modeling. Similar to searching by document, here we also provide the relevant detailed snippets for each document, like the interface shown in Figure 5(c). A set of relevant datasets for the queried dataset are also returned to the users. Here the relevance between two datasets is defined as the number of common documents that refer to both datasets. This can help users to better understand the relationships between different datasets. Figure 5(b) shows the relevant datasets for "movielens", including *IMDB*, *EACHMOVIE*, and *EPINION*.

Dataset Exploration. Finally, we allow further exploration of the connections between datasets and research papers. When a user is browsing the dataset of some document in Figure 5(a), they can focus on this dataset by clicking the `Explore` button to see which other papers used it. This switches to the `Search by Dataset` interface, as shown in Figure 5(b). Likewise, when a user is looking through the documents with respect to some specific dataset in Figure 5(b), they can easily locate the other datasets used in the document by clicking the `Explore` button at the end of the title line. In this way, users are able to explore the datasets broadly and deeply by following the dataset and document relationship links found by the system.

IV. Future Work

There are many ways in which we plan to extend the functionality of our system. As noted at the start, academic search engines extract metadata, in the form of author lists, year of publication, venue and so on. A first step will be to present this information for each paper in the system. But beyond this, it will be helpful to apply this information to datasets, to identify which authors and venues have used a dataset, and how its usage has varied over time. We can also look to the users of the system to help improve the search quality, by providing feedback on the extracted links, indicating errors, and identifying datasets within papers not identified by the classifier. With such participation, the precision, recall and coverage of the system can be further improved.

References

[1] Apache PDFBox. `http://pdfbox.apache.org/`.
[2] R. Bunescu and R. Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the Association of Computational Linguistics*, 2007.
[3] P. Haffner. Scaling large margin classifiers for spoken language understanding. *Speech Communication*, 48(iv):239–261, 2006.
[4] R. Jones, R. Ghani, T. Mitchell, and E. Rilo. Active learning for information extraction with multiple view feature sets. In *Proceedings of the ECML Workshop on Adaptive Text Extraction and Mining*, 2003.
[5] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.