JSTOR - Data for Research

John Burns, Alan Brenner, Keith Kiser, Michael Krot, Clare Llewellyn, and Ronald Snyder

301 E. Liberty, Ste. 330,
Ann Arbor, MI 48104
USA
{john.burns,alan.brenner,keith.kiser,michael.krot,
clare.llewllyn,ronald.snyder}@ithaka.org

Abstract. JSTOR is a not for profit organization dedicated to helping the scholarly community discover, use and build upon a large range of intellectual content in a trusted digital archive. JSTOR has created a new tool called "Data for Research" that allows users to interact with the corpus in new ways. Using DfR researchers can now explore the content visually, analyze the text and the references, and download complex datasets for offline analysis.

Keywords: JSTOR, text analysis, data, research, users, dataset, corpus.

1 Introduction

In pursuit of an overarching goal of technological innovation [1, 2], JSTOR is working in close collaboration with other researchers throughout the academic community to build new tools in order to enhance, explore, and allow more effective use of the content in the archive. This will bring significant additional value to the communities that JSTOR serves. Data for Research (DfR) [3] is a service that provides a web based visualization tool to explore and analyze the data within JSTOR and the ability to create, refine and download datasets of metadata, word frequency counts, references, n-grams and key words.

2 Visualizing JSTOR

A web interface allows users to explore content in the archive, discover features of interest, and ultimately generate downloadable datasets. Along with full-text searching of all content, the tool also provides faceting by discipline, journal, publisher, author, language, reviewed work/author, article type, the presence of references, time span or specific date. To assist in analysis of the data, graphs are employed to highlight key aspects about the slice of the JSTOR corpus the user has selected. These graphs include; frequency of articles returned by year, relative weight of articles returned adjusted to reflect total number of articles held in JSTOR by year, and distribution of articles across disciplines (Fig. 1). Furthermore, users are presented with total number of articles available in each facet for subsequent dataset refinement.

M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 416–419, 2009.

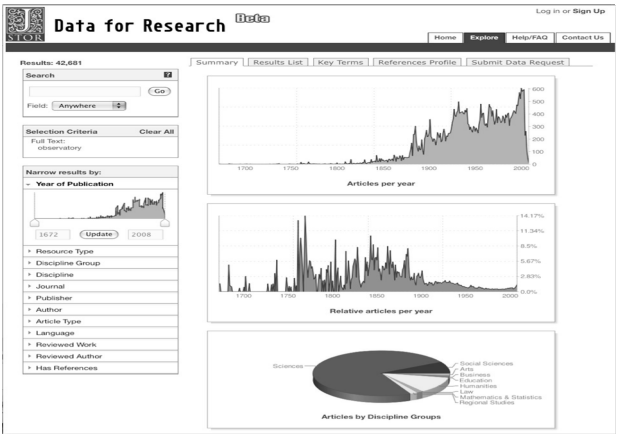

Fig. 1. Data for Research Summary view with a full text search for the word "observatory"

In addition to graphing tools, users have access to individual article metadata, key terms, references and n-grams. Key terms for the entire dataset (generated using TF/IDF [4]) are displayed as tag cloud where word size is in relative proportion to term weight. The tool also provides a dataset level view of references with a textual summary of references in and charts that show average number of references and average age of references by year (Fig. 2).

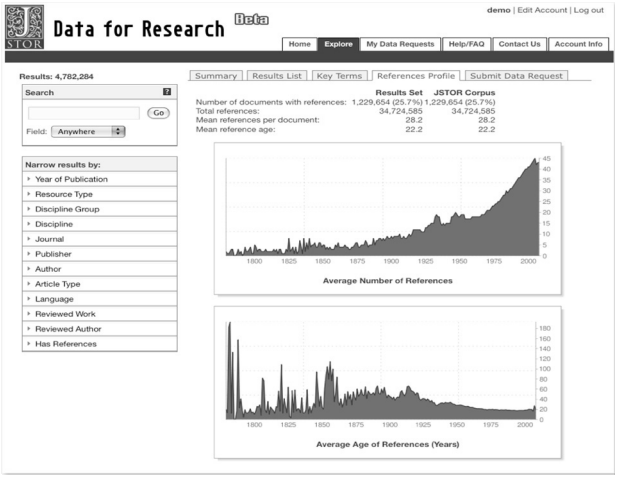

Fig. 2. Data for Research Summary Reference View for the entire archive

3 Creating a Dataset

Once users have defined a dataset they can choose to download it one of several data types (Table 1) in either CSV or XML formats. After the dataset is generated, users receive an email with a link that allows them to download the files and perform offline analysis.

Table 1. Available Data Types

Type	Description
Word Counts	Word occurrence frequency counts
Citations	Basic citation metadata
Bigrams	Two-word sequences that occur consecutively
Trigrams	Three-word sequences that occur consecutively
Quadgrams	Four-word sequences that occur consecutively
Key Terms	Auto extracted key terms (TF-IDF)
References	A list of works cited by articles in the data set

Optionally, users interested in programmatic access can use a combination of SRU (Search and Retrieve via URL) [5] and CQL (Context Query Language) [6] to download data directly. The DfR SRU service is found at http://dfr.jstor.org/sru. Pointing a browser at this address will produce a web form for defining CQL queries. The data returned by an SRU *searchRetrieve* query contains basic bibliographic data such as a unique document identifier, the document title, author names, publisher, and date of publication. The document identifier can then be used to download all data types from Table 1 in an XML format.

4 System Architecture

Various pieces of readily available, open-source software have been integrated atop the JSTOR corpus to form the Data for Research analysis framework. At the time of writing, JSTOR is home to nearly 5 million articles from over 1300 journals in over 50 languages spread across nearly 350 years. Given the very large amount of data, a chief requirement of any software was that it continues to perform at a large scale.

DfR employs the web framework Django for user account and permission management as well as the front-end interfaces [7]. Searching and faceting work by storing article data in a Lucene [8] index in conjunction with the search server Solr [9]. In addition to these, for speed and simplicity, a Berkeley Database [10] is used to store word counts and n-grams used in dataset production. Custom Java and Python code are also used throughout the data the application.

5 User Need

Throughout the years, JSTOR has received many requests for data from researchers from a wide variety of scholarly disciplines and with varying degrees of technical ability. Before the Data for Research tool, each individual request required a response to be hand crafted. Needless to say, this was inefficient, time consuming, and often resulted in long waiting periods for the end user. This experience informed development of the Data for Research tool.

Initial observations about users of the Data for Research site indeed confirm a large variety in our user base, many of whom have helped inform feature development. Since its inception, interest has been strong from all segments of the scholarly

community. Further functionalities have been added when they were requested via feedback from the site. Linguists requested the n-gram functionality, various groups requested access to reference information, and the API was developed through close consultation with users from the University of California Digital Library who expressed an interest in having such a tool.

6 Example Analysis

Initial analysis of the JSTOR corpus using DfR has highlighted some interesting case studies that may provide a simple illustration of how the system can be used:

- The long s – the character found extensively in pre-1800 documents that looks like an "f" – rapidly dropped out of use after 1800.
- Use of the word "hath" dies out of general use around 1900. Hath is only used in language and literature and historical journals after 1900.
- Searching for the word *chymistry*, gives a line graph that demonstrates how usage of the word fading as time increases, until suddenly at the turn of this century there is a spike in usage. Analysis shows the occurrence of this word is in citations – modern scholars are using digitized works on the Internet in their research.

References

1. JSTOR, http://www.jstor.org
2. Schonfeld, R.: JSTOR a History. Princeton University Press, Princeton (2003)
3. Data for Research, http://dfr.jstor.org
4. Salton, G., Buckley, C.: Term-weighted approaches in automatic text retrieval. Information Processing & Management (1988)
5. SRU: Search/Retrieval via URL, http://www.loc.gov/standards/sru/
6. CQL: The Context Query Language, http://www.loc.gov/standards/sru/specs/cql.html
7. Django, http://djangoproject.com
8. Lucene, http://lucene.apache.org/
9. Solr, http://lucene.apache.org/solr/
10. Yadava, A.: The Berkeley DB Book. Apress (2007)