Slashpack: An Integrated Tool for Gathering and Managing Hypertext Data

Christopher H. Brooks, Monica Agarwal, Jason Endo, Ryan King, Nancy Montanez and **Rudd Stevens**
Computer Science Department
University of San Francisco
2130 Fulton St.
San Francisco, CA 94117-1080
cbrooks@cs.usfca.edu
http://www.cs.usfca.edu/∼brooks

Abstract

Many interesting Web-based AI problems require the ability to collect, store and process large text datasets. To address this problem, we have developed Slashpack, an integrated toolkit for collecting and managing hypertext data. Currently, we are using Slashpack to study the effectiveness of tagging as a mechanism for organizing and searching blogs, and also to study community structure in the blogosphere.

Introduction

One feature of next-generation Web technologies is the focus on user-centered creation, such as blogs. With the advent of decentralized creation, modification, and annotation of information comes the need for tools that help both users and researchers to sort, filter, and make sense of this information. Given the vast amount of information available, automated techniques are needed to help people with these problems. Our current research questions study the ways in which users can more easily discover, describe and share information.

A necessary precursor to this research, as well as a useful resource for the community at large, has been the development of a generic toolkit that will allow us to gather and manage large amounts of hypertext data. This toolkit, named Slashpack, has been built to collect, store, and process data from a variety of different sources. We are making Slashpack available to the general research community for researchers interested in working with relatively large (50-100 GB) text collections.

Our current research involves using Slashpack to study two problems related to the blogosphere: the effectiveness of tags as a mechanism for categorizing and retrieving information, and the identification of authorities and agenda setters within the blogosphere.

Architecture

Slashpack is composed of three parts: the collector, the storage manager, and the processor. The architecture is shown in Figure 1 and discussed below.

Collector

The first component in Slashpack is the *collector*. The collector is designed to crawl a diverse set of data sources, gathering all of the hypertext data in that source. Sources include the Web (including the ability to crawl specific sites and domains), local file systems, ZIP and tar files, and third-party data sources such as Technorati, via their Web Services API.

When crawling the Web, the collector begins with a set of seed pages and a set of regular expressions that describe URLs that should be included in the crawl. As a document is collected, its outward links are extracted and placed in a queue for later extraction. MD5 hashes of each document are kept to avoid duplication. The collector also employs a 'niceness' parameter, which allows the user to control how often an HTTP request is sent to a particular web server. The collector also includes its own DNS cache, as well as a hand-tuned robots.txt parser, which speeds up the collection phase significantly.

Storage Manager

As the collector harvests documents, it captures relevant metadata about each document, such as its URL, date collected, size, tags, outward links, and mimetype. This metadata is passed on to the Storage Manager.

The storage manager writes the XML metadata into a SQL database and stores the data itself in the filesystem. This allows us to perform SQL queries on the metadata, while still allowing fast retrieval of the data itself.

Processor

Once the data has been archived by the Storage Manager, a user-defined pluggable set of filters are applied. Among other tasks, these filters can strip HTML tags, remove stop words, classify words according to part of speech, perform frequency counts, and classify pages according to language.

Applications

Slashpack is primarily a means to an end. While it is interesting and useful to build a general-purpose toolkit for managing and processing hypertext datasets, we are primarily interested in using Slashpack to answer interesting questions about how users share and describe information. Currently,

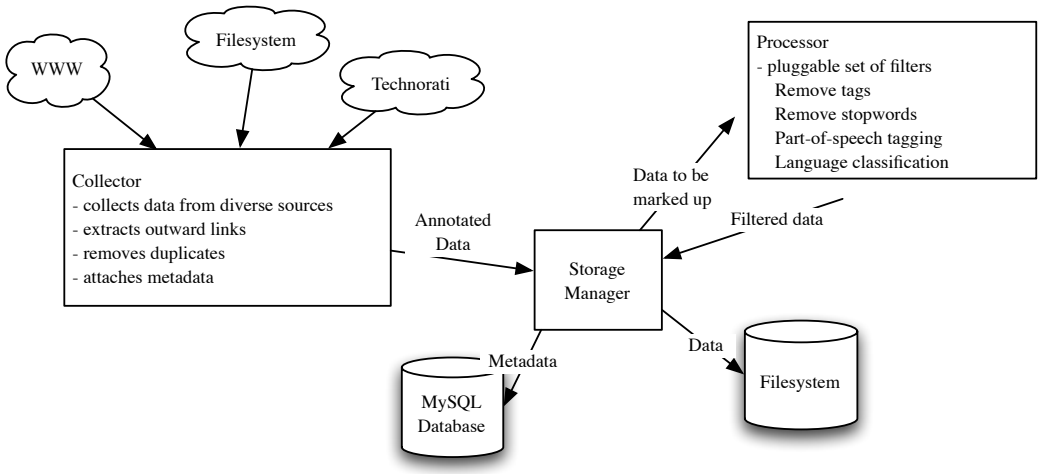

Figure 1: The Slashpack Architecture. The collector gathers data from one or more sources, and passes the data to the collector. The metadata is stored in a MySQL database, and the data written to the filesystem. The processor then filters the data and adds additional application-specific metadata into the database.

we are using Slashpack to investigate the shared annotation and propagation of information in the blogosphere.

Blog data has a great deal in common structurally with normal Web data, but there are also some differences. A blog entry often has additional contextual information, including the blog it is contained in, trackbacks, and author-assigned tags. All of this must be collected and stored by Slashpack. The current Slashpack architecture makes this easy; the user writes a Protocol Handler that specifies how to interact with the data source and what data to extract, as well as a Storage Manager component that indicates the structure of the SQL tables used to hold the metadata.

Tag Analysis

One thread of our work on shared information spaces has focused on the use of tags to categorize and organize blog entries (Brooks & Montanez 2006). Tags are keywords that can be assigned by authors or readers of a document as a way of summarizing its content. What is novel about tags is their ability to be shared; that is, one user can see the tags that other users have assigned to a document. Tagging proponents argue that this leads to a decentralized, usage-based determination of meaning known as *folksonomy*.

We quantitatively analyze blog entries that share tags to determine whether tags are useful as a means of enhancing search and information retrieval. Clusters of articles that share tags are measured for pairwise similarity and compared to randomly selected clusters, as well as clusters of articles that share automatically extracted relevant keywords. We show that tags are most effective at grouping articles into large, broad categories such as 'Food' or 'Games' and, contrary to popular opinion, are not very effective at annotating articles for search and retrieval tasks. We show that using standard IR techniques to automatically extract relevant words produces tags that are more effective in search tasks. The primary weakness of tag-based systems is the fact that tags are propositional entities. Users are unable to specify relationships between tags, group them into subclasses or superclasses, or otherwise reason about their semantics. We argue that a more expressive tagging language that allows users to specify these sorts of relationships is needed, albeit one that retains the ease of use of current tagging systems. We also show how a hierarchy of tags that shares characteristics with hand-labeled hierarchies can be automatically generated using unsupervised clustering methods. This provides some evidence that AI and IR techniques can be combined with tagging to provide additional annotated semantics without additional user burden.

Community Structure

We are also currently using Slashpack to develop techniques for identifying communities, topical authorities and "agenda setters" within the blogosphere. For example, within the political blogosphere, we are interested in determining which bloggers link to each other, which bloggers are responsible for popularizing phrases and stories that make it into the blogosphere more generally, and how network structure affects the dispersal of information. This follows on the work originally done by Adamic and Glance (Adamic & Glance 2005).

References

Adamic, L. A., and Glance, N. 2005. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of LinkKDD-2005*.

Brooks, C. H., and Montanez, N. 2006. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th International World Wide Web Conference (WWW-2006)*. Edinburgh, UK: ACM Press.