PrEV: Preservation Explorer and Vault for Web 2.0 User-Generated Content*

Anqi Cui[1], Liner Yang[1], Dejun Hou[2],
Min-Yen Kan[2], Yiqun Liu[1], Min Zhang[1], and Shaoping Ma[1]

[1] State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
{cuianqi,lineryang}@gmail.com, {yiqunliu,z-m,msp}@tsinghua.edu.cn
[2] School of Computing, National University of Singapore, Singapore
houdejun214@gmail.com, kanmy@comp.nus.edu.sg

Abstract. We present the **Pr**eservation **E**xplorer and **V**ault (*PrEV*) system, a city-centric multilingual digital library that archives and makes available Web 2.0 resources, and aims to store a comprehensive record of what urban lifestyle is like. To match the current state of the digital environment, a key architectural design choice in *PrEV* is to archive not only Web 1.0 web pages, but also Web 2.0 multilingual resources that include multimedia, real-time microblog content, as well as mobile application descriptions (e.g., iPhone app) in a collaborative manner. *PrEV* performs the preservation of such resources for posterity, and makes them available for programmatic retrieval by third party agents, and for exploration by scholars with its user interface.

Keywords: Preservation, Archive Visualization, API, Web 2.0, User-Generated Content, NExT, PrEV.

1 Introduction

Not long ago, the Web was a largely homogenous digital environment, with web servers serving static authored web pages, enriched by embedded image, audio and video resources. We consumed these resources as readers, and our indexers and archivers did the same. Archiving such content, while difficult due to scale and the necessary curation, was otherwise technically feasible. This type of initiative is exemplified by the Internet Archive's Wayback Machine[1], which provides a URL-based navigation on web pages. Through a calendar interface,

* This work was supported by Natural Science Foundation (60903107, 61073071), National High Technology Research and Development (863) Program (2011AA01A207) and the Research Fund for the Doctoral Program of Higher Education of China (20090002120005). This work has been done at the NUS–Tsinghua EXtreme search centre (NExT).

[1] http://www.archive.org/web/web.php

P. Zaphiris et al. (Eds.): TPDL 2012, LNCS 7489, pp. 101–112, 2012.

a scholar can view many archived instances of specific web pages, reaching back as far as the mid 1990s.

Fast forward to today's Web, a web that is centered on a new form of content: User-Generated Content (UGC). Such content is tacked on at the end of Web 1.0 pages (e.g., news articles with commenting) or at centralized (e.g., restaurant reviews on Yelp) or decentralized (e.g., personal blog sites running WordPress) Web sites. It comprises people's opinions and comments and is updated more frequently than the near-static Web 1.0 pages. Today's web is now much more interactive than before, better catered to the spectrum of devices that we use to consume digital resources now.

This interactivity has come with a cost: the Web has become more fragmented and harder to archive. Popular social media sites must restrict access to sensitive personal content. Many pages are dynamically created via push technologies, making simple crawed versions of pages largely devoid of content. Smartphone and tablet applications ("app") make up a large percentage of consumed bandwidth, but accessing information about these apps is restricted to their proprietary devices and their communication protocols. With the dynamicism of Web 2.0, we may only be able to capture a particular user experience – as how a website looks might change from user to user, instant to instant[2]. Clearly, to archive the spectrum of UGC that collectively represent today's Web is more challenging.

However, it is a challenge that we must rise to, in order to paint a holistic picture of life today for future generations to appreciate. A key observation we make is that such an archiving initiative must present these myriad resources in a unified manner. If we only archive piecemeal, the future scholar may make erroneous conclusions based on his incomplete picture of our lives. To surmount the challenge of the archiving the Web 2.0 spectrum, we scale back the scope of the resources that we archive. Our project, the **Pr**eservation **E**xplorer and **V**ault, hereafter *PrEV*[3] is currently fielded to archive only content related to the two capital cities of Beijing and Singapore, in both Chinese and English.

In preserving content for posterity, we must amass a large focused collection of resources that are valuable and of potential interest to developers looking to harvest and extract information from today's web. For this reason, we make another key decision to opportunistically support third-party programming data access.

We use two running scenarios to motivate the approach and architecture taken with *PrEV*:

(1) Suppose many years later, Ryan, a secondary student is doing his term project about Singaporean hawker center (cooked food center) history, He aims to obtain some historical pictures of people and food at hawker centers, but finds only secondary texts that mention what it was like to eat a meal at the

[2] `http://blog.dshr.org/2012/05/harvesting-and-preserving-future-web.html`

[3] So named to contrast with its umbrella project name – NExT: the NUS–Tsinghua EXtreme search centre.

hawker center. They contain digested accounts, and do not have the raw data and primary source, first-hand data that Ryan needs for his project.

PrEV addresses this problem by archiving pictures of Singapore (including hawker center experiences), associating them with the time and place where they were taken. In encountering *PrEV*, Ryan issues the simple query "hawker", which lists all the relevant records in the history (Fig. 1, left). Ryan picks out the picture records in different time points. Ryan uses the word cloud summary feature, (Fig. 1, right) highlighting the most frequent terms found in the records relevant to the query. Browsing the data, he generates several more word clouds for different periods in the collection, discovering salient aspects of hawker centers, including locations (*bedok*, *toa payoh*), and cuisine (*chicken rice*). From these records, he issues subsequent queries to *PrEV*, browsing through to find related blog, forum and Twitter posts that give Ryan a feel for the old-style dishes and environment that made up the hawker center environment of 2012.

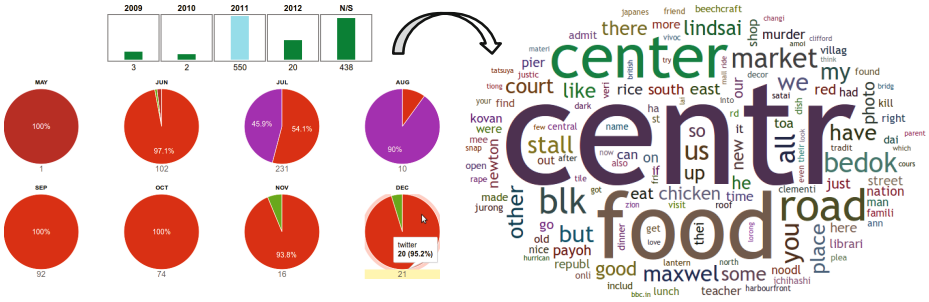

Fig. 1. *PrEV* records with respect to "hawker". Left: collection statistics with respect to the query; Right: a (stemmed) word cloud summary restricted to microblog data from Twitter and Weibo (a Chinese microblogging service), and adding Flickr records.

(2) A startup company, Blueberry, aims to develop their "Follow Me" iPhone travel guide application for Beijing and Singapore. Before they start, the design team wants to review the existing competitor apps to understand their weaknesses through the user's comments. However, they have to read through competitor app descriptions and reviews manually, as the data in the Apple App Store can only be accessed through an iPhone or iPad and not by any server based system.

Later, the team comes across *PrEV*, which archives this data of interest; in particular, descriptions and reviews of mobile apps about both cities.. They contact *PrEV*'s administrators and obtain an API key for retrieving the large amount of archived data programmatically. They query *PrEV* using a REST-compliant syntax and receive the results in JSON format, and are able to see how the application's descriptions and popularity changed over time.

Both scenarios are largely supported by our current implemented version of *PrEV*. Underlying the scenarios is our assumption that users want to retrieve the

```
{ "total":86, "count":10, "totalpage":9, "page":0, "data":[
  { "crawlresource":"twitter", "encoding":"en", "tweetcreatedat":"Sun Oct 02 19:56:51 +0800 2011",
    "url":"http://twitter.com/#!/ChristianLeeVO/status/120467276104339457",
    "maincontent":"Check this video I shot of Pulau Ubin for our Travel Now Singapore webseries and iphone
app http://t.co/mvfgJDqP via @youtube" },
  { "crawlresource":"weibo", "encoding":"zh", "weibocreatedat":"Sat Jan 14 19:13:27 +0800 2012",
    "url":"http://www.weibo.com/1910529591/y0Lf5mFAk",
    "maincontent":"#App推荐# 旅程规划：Routes. Planning your journeys【出行必备】，iPad/ iPhone通用。这款应用
可让你规划旅游景点，像是到某景点去拜访或是去某家大卖场购物等等。它会算出需要多远的距离以及所需的时间，就像...
http://t.cn/z0gtfEr （分享自 @App每日推送）" },
  { "crawlresource":"sgbjapps", "encoding": "others", "crawltime":"Wed Dec 23 00:00:00 +0800 2009",
    "maincontent":"Do You Love Travel ? If Yes, You Should Not Miss This App. Updated For Now! Download
this app to your iPhone to enjoy these beautiful scenery anywhere you go! These pictures are HD Photo You
can download the image to your iPhone or iPod and make it to wallpaper. No Ad No Wifi!",
    "name":"A Tourist Paradise <Singapore>" }, ... ] }
```


Fig. 2. Some actual multilingual *PrEV* data relevant to the travel application domain. Results subsampled to highlight the variety of sources (Twitter, Weibo and App Store).

historical records on some topic (e.g. an entity, a product), filtering the content not only by keyword but by other facets such as time and data source.

We present *PrEV*'s architecture and implementation in the remainder of this paper. As the scenarios illustated, *PrEV* has three main missions: 1) to archive the myriad Web 2.0 about cities, 2) provide them in an exploratory browsing interface, as well as 3) providing them in a programmatic interface. To achieve this, *PrEV* uses a three-layer framework: 1) a preservation layer to store the different types of records in a unified structure, 2) an indexing layer that allows fast retrieval on different facets, and 3) an interface for presenting the data to both browsing users and programmatic agents. The loosely coupled design makes it possible to preserve both text and multimedia records together, as well as to provide retrieval and visualization from different views.

2 Related Work

There is much work relating to the wider aspects of digital preservation. We limit our discussion to the most relevant work on archiving Web data and user-generated content. We also briefly review the user interfaces and visualization techniques that have been used to explore such archives.

Web Archiving. The preservation of the Web has been an issue of interest early, as the Web became the method of choice to disseminate information. Research topics include crawling methodologies, version control, recovery of broken links, among other topics. A seminal system that continues to operate today is the Internet Archive's Wayback Machine [14], which takes a broad approach to archive historical web pages by URL, collecting multiple snapshots of websites. Their effort has been a reference for many succeeding studies, including country-restricted web archives in Norway [2] and China [24], among other national initiatives. The International Internet Preservation Consortium (IIPC) [8] associates libraries in different countries to preserve the Interent contents (mainly web pages) crawled by each library themselves. While some relevant work describes how to make curation decisions in Web archiving to focus web

crawling efforts, in *PrEV* we take a broad approach, collecting and storing any data provided by trusted third parties.

Some of the studies on web preservation face some technical challenges of data format standard, storage safety, scaling issues or selection priorities [3,10,19,20]. In contrast, our focus differs in our system objective, i.e. to provide access to Web data, including but not limited to traditional Web pages. Our challenges center around the organization of the variegated data types that we collect, and ensuring usable access to the data in a unified interface.

Another web archiving research focus is to discover and restore access to deleted or missing pages [2,5,11,17,22]. To support this, up-to-date crawls and rate of change estimation are a necessity, as pages change constantly. This phenomonon is exacerbated in the scope of *PrEV*'s Web 2.0 data, in which UGC are often short but updated frequently [3].

Separately, proper data access is also a concern in shared archives that span users from multiple institutions or organizations [4]. Such archives may have to meet different requirements in data access and sharing [9]. Specific infrastructures have been designed for such multi-level access control. Some studies design a multi-layer architecture, with different layers geared towards handling data preservation, indexing or search access [11]. Since our system is a public data archive, we have fewer restrictions on our data, but we ingest data from certain sources that have restrictions (e.g., Twitter). We adopt a similar multi-layer architecture in *PrEV*, loosely coupling different functionalities as serial layers.

Web 2.0 Archiving. Web 2.0 is about user-generated content. This bottom-up flow – from users to website – brings more challenges for preservation [7,23]. The multimedia, real-time and streaming UGC are usually difficult to crawl with traditional techniques. Many Web 2.0 pages' ultimate appearance within a user's web browser conditions not on just the web page but external cascaded style sheets, embedded applets, scripts, and more recently iframe contents and dynamic content written by push technologies (i.e., AJAX/XHR). More efficient headless browsers that simulate the actual rendered page and execute the embedded scripts are needed to crawl and preserve such contents.

While the call to arms to preserve UGC is widely known, UGC are individually the focus of different research groups. Twitter is perhaps the UGC source with the most active archiving movement. [15,18,25] archive Twitter data for their own analyses. Existing commercial websites also provide access to the resyndicated Twitter archive, such as *Topsy*[4], *TwimeMachine*[5] and *indextank*[6]. They offer keyword-based retrieval with different facets (time, language, etc.) on the Twitter messages. Similarly, researchers working on analyzing Flickr images, YouTube videos, Yelp reviews are all crawling these sites individually. Currently, there is no unified platform for researchers or users to obtain a holistic view that cuts across UGC sources, to search through all these UGCs, even for limited topics.

[4] `http://www.topsy.com/`
[5] `http://www.twimemachine.com/`
[6] `http://www.indextank.com/`

Within the realm of private social network data, less work has been done. A notable exception, is McGown and Nelson's work [16] which developed a browser extension to back up a user's Facebook presence. Greplin[7] takes a similar approach, emcompassing multiple social network sites. Both embody personal backup solutions, but not large-scale preservation.

In contrast to all of these works, *PrEV* aims to un-silo these Web 2.0 resources and archive them together.

Web History Visualization. Once the Web data is crawled and archived, how can it be effectively presented to the end user? Two web page preservation systems, the Internet Archive and the Memento Project [21] present the historical page, given a manually-specific date. While fine for traditional Web 1.0 web pages, it may not make sense for UGC since they are often spread over multiple 2.0 sites. On the other hand, changes between different versions of a page are not presented directly. In most studies, changes are measured with respect to text (keywords, tags, etc.) [1,6,12,13]. In a typical change visualization, different versions of the page or relevant change areas are listed in columns, then related terms in the adjacent versions are connected with lines to show the change flow. Currently in *PrEV*, we are working towards defining what constitutes change. We present a stationary summary of a set of resources, via a *word cloud* generated from different resources, instead of simply comparing different versions.

3 System Architecture

The *Preservation Explorer and Vault* involves multiple data contributors, while serving the data to end users through both Web and API service endpoints. To achieve a loosely coupled system, we divide the system into three layers, as shown in Fig. 3. The three layers operated independently from each other, although some layers operate concurrently on individual machines.

1. *The Preservation Layer* interacts with the crawlers and handles long term storage. It reads the raw, crawled data and archives them in the file system or database permanently.
2. *The Indexing Layer* enables the necessary retrieval functionality of frontend service endpoints. Each archived record is processed into an indexable version, so that users can retrieve them by both content (keyword) and different facets (metadata). Multiple levels of indexing and processing can be run to generate different levels of automated analyses on the raw archived data.
3. *The Interface Layer* serves the data to end users. The interface modality varies based on the user requirements: e.g., content-based queries, visualizations or command-line access.

While the multi-layer design nicely modularizes the responsibilities at each layer, the interface between layers is the challenge. Issues that have been addressed include specifying data formats between layers, and lowering the latency between the initial crawl and eventual availability in the system. We now give a detailed view of each layer.

[7] `http://www.greplin.com/`

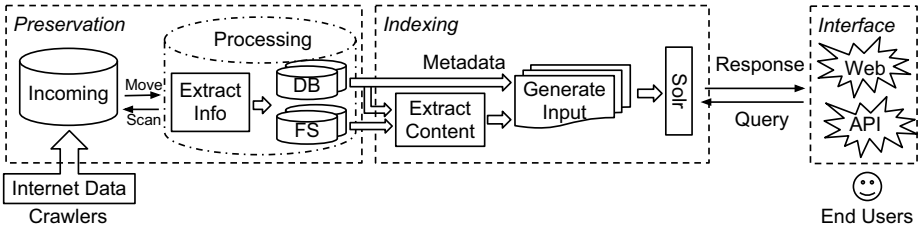

Fig. 3. *PrEV*'s overall three-layer system architecture

3.1 The Preservation Layer

At this first layer, *PrEV* collects data from different crawlers.

PrEV is a central repository project that binds several city-centric search research projects together. Staff and students spread across two institutions, Tsinghua University in China and the National University of Singapore in Singapore, are involved, and many of them run their own crawlers to fetch Web 2.0 data sources for their individual projects. Some of them crawl resources with respect to their own city, while some others crawl global resources. Our project mandates that they share their trusted crawled data with our central project. Up till now, we have been collecting multilingual data of city lifestyles from crawlers covering more than 300 million records (shown in Table 1). Though most of the resources are static (such as microblogs or news articles), dynamic resources (forum posts, product reviews, app data etc.) are gradually becoming a larger percentage as these 2.0 data sources are re-crawled more frequently.

A series of steps from collecting to storing are demonstrated below.

Incoming Data Detection. Our preservation process is architected as a federation of independent crawlers who push data to the central repository at their convenience. The crawlers, run by individual researchers, use the *PrEV* master server as the single point of contact to upload data to. To be clear, our system does not crawl resources but collects data from the crawlers.

Each crawler is registered as a user on the *PrEV* server. The master server also has a *PrEV* user that houses a writable directory, in which other users can write to. Crawlers can copy their crawled data to the server at any time. Crawlers also create a zero-sized *unlock* file, after a transfer is finished, to denote the associated file was successfully stored on the server. The *PrEV* user periodically scans the common directories for new files and their associated unlock files, and moves the files to a staging area for processing. In this collaborative way, different resources are united into the central server.

Sources from the individual crawlers are independent between each other. Our central repository benefits from this strategy, that requests from different locations may provide different views (responses) of a same resource at one time. Therefore, duplicate resources from different crawlers are all kept in the system; duplicate handling is the province of the indexing and presentation layers.

Table 1. City lifestyle data resources from multiple crawlers (as of May 2012)

Data Type	Resource	No. of Records
Microblog messages	Twitter	229 M
	Weibo	139 M
Photos with texts	Flickr, Panoramio	2 M
Food forum restaurants	27 Singapore sites	6 M (pages)
	Fantong, Dianping (Chinese)	78 K
Public forum posts	4 Chinese forums	1 M (approx.)
Product review products	7 e-commerce sites	70 K
News articles	Sina News	224 K
	Guardian, Channel NewsAsia, Skysports, CNN, Economist, FoxNews, NewYorkTimes, StraitsTimes	59 K
Wiki articles	Hudong (Chinese)	1 M
Traffic records	Singapore	24 K
	Beijing	19 K
Question Answering articles	Baidu Zhidao (Chinese)	33 K
	Yahoo! Answers, WikiAnswers	52 K
Mobile Apps	US App Store	617 K
	Android Market	345 K
	Blackberry, Windows	162 K

Data Format Recognition. Due to the diversity of data that *PrEV* archives, we allow several different formats for crawler submission. A microblog post only consists of a short text string, while a photo incorporates megabytes of binary data and text for its description and comments. We use three simple submission data formats to reduce the overhead for the staff maintaining the crawlers.

1. *Short text* data (e.g., microblog entries) are stored in a text file, i.e. each line of the file is considered as a single record. The meta-information is provided within the file.
2. *Single record files* (e.g., image files) are a raw, binary source of the record. Any record metadata is provided in a separate description file that accompanies the submission.
3. *Multiple record files* (e.g., web page archive) allow crawlers to compress and store multiple single raw files together, In this case, we consider the file as a concatenation of the records. A crawler must also provide the offset and size of each record, for extracting the corresponding record from the file, in a separate description file.

In the latter two cases, the raw and README files are zipped into a single file for upload. The zipped file is later extracted for content analysis and retrieval. This strategy also makes it possible to consider multiple files as one single record. For example, the source code (text) of a web page and its associated images and CSS style files are considered as a complete snapshot of the web page. Note that some crawlers only crawl texts or even ignore the header information of a web page. For this reason, we do not explicitly force crawlers to archive in well-known formats such as the Web Archiving File Format (WARC), although they may.

Record Storage and Backup. After extracting each record from the uploaded file, *PrEV* stores each record's metadata information in the database for management; any raw files are stored as simple files on a distributed file system.

The database is periodically backed up in a RAID 1 system (fully redundant mirror), while the raw files are stored on a RAID 6 file systems, able to withstand two simultaneous disk failures.

3.2 The Indexing Layer

A separate indexing process enables faceted navigation on the metadata stored in the databases. We use a single installment of Apache Solr[8] as our indexing service. It provides an HTTP-based method for data injection, in which the data to be indexed are submitted to a web service. In this way, we implement incremental indexing (while not particularly efficient). The workflow of the indexing system is shown in Fig. 3:

1. Fetch new records from the preservation layer.
2. For each record, extract the facets to be indexed based on its type. For example, *PrEV* extracts the text content of the microblog messages, descriptions of the photos, body text of the web pages. Other facets include the resource, data format, crawled time, record ID, etc. Facets are defined per-resource type.
3. The extracted fields are submitted for injection into the Solr server. After indexing, the corresponding record in the preservation layer is marked as "indexed", to prevent multiple indexing instances.

We use dynamic fields (supported by Solr and its underlying Lucene search core) to handle specific facets associated with certain data types. These fields are used for specialized query and presentation. For example, the *author* of a microblog post may be helpful to end users in defining their search scope, but authors are not generally attributed with all records that *PrEV* indexes. Therefore, for the microblog data type, we define the author's screen name and profile image as two additional dynamic fields, which can be used to retrieve a certain user's posts. The Solr `traverse` process is executed periodically to add the latest records to the index.

3.3 The Interface Layer

The interface layer currently has two service endpoints, which we described through the earlier scenarios. These are a web frontend for individual users and an API frontend for enterprise-level use.

Web Frontend. In the first scenario, Ryan asks *PrEV* to provide him the statistics on relevant records, as well as a summary of the text contents (Fig. 1).

With the help of the indexing layer, the web frontend issues a number of database queries to provide a calendar view of the number of matching records. The calendar view is hierarchical, allowing results to be drilled down from a year view to months, and to individual days, providing both macro and micro views

[8] `http://lucene.apache.org/solr/`

of the data. We show the proportions of different data types as dynamically-generated pie charts drawn via the Google Chart API. The page is implemented with AJAX to improve the user experience.

Results at any level can be used to form a word cloud, to get a feel for the individual resources at each level. The word cloud is dynamically generated, based on a random sample of records in the relevant range (usually a keyword + time range).

API Frontend. Enterprise-level users, as in the second scenario, usually process a much larger number of records at once, requiring a batch mechanism to retrieve data. *PrEV* provides a RESTful API service that implements an API which includes user-level authentication to achieve this. The functions implement facet search, specified in parameters, to access the data in a flexible manner. For example, users may choose the data containing some query from certain resources within a specified time range.

On the *PrEV* website, we created a forum that combines user management and API registration. In addition to the standard troubleshooting and broadcast use of such a forum, a forum user – with appropriate permission – is issued a standard API key for authenticated API requests. The API key is assigned after the user registration is approved in the forum.

The API uses user-level authentication and performs two services: rate limiting and data access management. Each API call needs to provide the API key in the request. The rate limiting ensures that users can only send up to their allowed quota of requests per hour. The data access management ensures that a user can only access the types of data he has been authorized for. We also created an API sandbox, to help familarize our users with the functionalities we provide in the programmatic API.

The workflow below demonstrates the steps from reading the request to generating the response:

1. The web service receive a URI as the request from the user. The URI must contain a parameter as the API key.
2. The system checks if the API key is valid. If so, it finds the corresponding user information, including his rate limiting level and data access level. Otherwise, the request is rejected via generation of an HTTP error.
3. The system checks the rate limiting counts. If the user has exceeded his current use quota, the request is rejected.
4. The system generates a Solr query based on the request type, user's data access level and the request parameters, and sends this to the Solr server.
5. The system reads the query results from the Solr server, then transforms it to the response format, and returns it to the end user. The response header contains rate limiting information, while the body contains the data.

The rate limiting counts of each user is reset per cycle. This strategy is used by most RESTful API websites (such as Twitter). Besides the header of the response, the user can access one certain API to query his rate limit status, or request for their accessible resources.

4 Conclusion and Future Work

We have presented *PrEV*, the Preservation Explorer and Vault. *PrEV* is a city-centric archiving system, modeled to archive and unify multilingual data as represented by the current Web 2.0 paradigm. Our indexing layer implements faceted search, allowing users to access the data in a flexible manner. This includes internal natural language processing engines, which freely access the raw and previously-processed archives to process and deposit back annotations on the material.

While still under development, we argue that our user-oriented interfaces and APIs already provide flexibility for both individual scholars looking to browse the archival data and enterprise-level automation that seek to programmatically access a large amount of the crawled data. In addition, we plan to continuously involve more resources such as geo-location based contents, personalized pages in different languages.

In future work, we plan to continue to improve system performance, and support more community standards (web archive access via Memento [21]) and conduct formal evaluations, while enhancing the user interfaces to support better visualization of the changes in the collection. We plan to implement comparative visualization that will complement the faceted aspects of the current *PrEV* collection.

Acknowledgments. We are indebted to the many students and staff who provided the crawlers that supply *PrEV* with its data. The NExT Search Centre is supported by the Singapore National Research Foundation and Interactive Digital Media R&D Program Office, MDA under research grant (WBS: R-252-300-001-490).

References

1. Adar, E., Dontcheva, M., Fogarty, J., Weld, D.: Zoetrope: Interacting with the ephemeral web. In: Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, pp. 239–248. ACM (2008)
2. Albertsen, K.: The paradigma web harvesting environment. In: Proceedings of the 3rd Workshop on Web Archives, pp. 49–62 (August 2003)
3. Ball, A.: Web archiving. Tech. rep., Digital Curation Centre, UKOLN, University of Bath (March 2010)
4. Campbell, L.E.: Recollection: Integrating Data through Access. In: Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) ECDL 2009. LNCS, vol. 5714, pp. 396–397. Springer, Heidelberg (2009)
5. Chang, H.: Enriched Content: Concept, Architecture, Implementation, and Applications. Ph.D. thesis, New York University (2003)
6. Collins, C., Viegas, F., Wattenberg, M.: Parallel tag clouds to explore and analyze faceted text corpora. In: IEEE Symposium on Visual Analytics Science and Technology, VAST 2009, pp. 91–98. IEEE (2009)
7. Dougherty, M., Meyer, E., Madsen, C., Van den Heuvel, C., Thomas, A., Wyatt, S.: Researcher engagement with web archives: State of the art (2010)

8. Hallgrímsson, T.: The International Internet Preservation Consortium (IIPC). In: Conference of Directors of National Libraries (CDNL 2005), Oslo, Norway, pp. 14–18 (2005)
9. Hockx-Yu, H.: The past issue of the web. In: Proceedings of the ACM WebSci Conference 2011, pp. 1–8 (2011)
10. Hodge, G.: An information life-cycle approach: Best practices for digital archiving. Journal of Electronic Publishing 5(4) (2000)
11. JaJa, J., Song, S.: Robust tools and services for long-term preservation of digital information. Library Trends 57(3) (2009)
12. Jatowt, A., Kawai, Y., Tanaka, K.: Visualizing historical content of web pages. In: Proceedings of the 17th International Conference on World Wide Web, pp. 1221–1222. ACM (2008)
13. Jatowt, A., Kawai, Y., Tanaka, K.: Page history explorer: Visualizing and comparing page histories. IEICE Transactions on Information and Systems 94(3), 564 (2011)
14. Kahle, B.: Preserving the Internet. Scientific American 276(3), 82–83 (1997)
15. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600. ACM (2010)
16. McCown, F., Nelson, M.: What happens when facebook is gone? In: Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 251–254. ACM (2009)
17. Nelson, M., McCown, F., Smith, J., Klein, M.: Using the web infrastructure to preserve web pages. International Journal on Digital Libraries 6(4), 327–349 (2007)
18. Petrovic, S., Osborne, M., Lavrenko, V.: The Edinburgh Twitter corpus. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pp. 25–26 (2010)
19. Ronald Jantz, M., Mlis, M.: Digital archiving and preservation: Technologies and processes for a trusted repository. Journal of Archival Organization 4(1-2), 193–213 (2007)
20. Seadle, M.: Selection for digital preservation. Library Hi Tech. 22(2), 119–121 (2004)
21. Van de Sompel, H., Nelson, M., Sanderson, R., Balakireva, L., Ainsworth, S., Shankar, H.: Memento: Time travel for the web. Arxiv preprint arxiv: 0911.1112 (2009)
22. Song, S.: Long-term information preservation and access. Ph.D. thesis, University of Maryland, College Park (2011)
23. Thomas, A., Meyer, E., Dougherty, M., Van den Heuvel, C., Madsen, C., Wyatt, S.: Researcher engagement with web archives: Challenges and opportunities for investment (2010)
24. Yan, H., Huang, L., Chen, C., Xie, Z.: A new data storage and service model of China web infomall. In: 8th European Conference on Research and Advanced Technologies for Digital Libraries The 4th International Web Archiving Workshop (IWAW 2004), Bath, UK (2004)
25. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the fourth ACM International Conference on Web Search and Data Mining, pp. 177–186. ACM (2011)