OLD DOMINION UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MASTER'S PROJECT REPORT

How I Changed Over Time: A webservice to summarize TimeMaps based on SimHashed HTML content

Author:

Maheedhar Gunnam

Advisor:

Dr. Michele C. Weigle

Dr. Michael L. Nelson

A report submitted in fulfillment of the requirements for the degree of Master In

Computer Science

2018

Acknowledgement

I would like to express my sincere gratitude to my project advisor Dr. Michele C. Weigle, Associate Professor, Department of Computer Science at Old Dominion University, for being a remarkable source of inspiration to me and for guiding me throughout the project. My love for visualization started during course Information Visualization in Spring 2017 under Dr. Weigle. Visual representations are always powerful and more effective when compared to other methods of representing data. I would like to thank Dr. Michael L. Nelson for co-advising efforts on this project. He has been very understanding and supportive at every step and hence, I could keep track of my progress and finish the project as intended.

I want to thank NEH ODH for the grant and their support, which made this project possible.

I take this opportunity to thank Sawood Alam, PhD Candidate, Web Science and Digital Libraries (WSDL) lab, Computer Science Department, Old Dominion University, for always being there to help me understand some aspects and spending time in making design decisions.

I would like to thank Miranda Smith for being very helpful throughout the semester working with me on this system, I want to thanks all other members from WSDL Research Group who provided me with related resources and important comments during my work with the Group. My special thanks to Surbhi Shankar Masters Graduate, Mat Kelly PhD Candidate and Dr. Ahmed Alsum who has done initial work of this project. Finally, my sincere thanks to the Department of Computer Science for providing me all the required resources and infrastructure to complete this project successfully.

Abstract

With the increase in the dynamic nature of the web, often the content of a web page grows, changes, and might be shrunk. And with these pages being archived numerous times, they serve as the digital history for those changes that are long gone from the live page. But visualizing over these numerous different archived copies, or mementos, with the intention of perceiving the major changes over time is nearly impossible, as the memento count can be very high. In case of cnn.com, the web page has been archived 188,966 times. This TimeMap summarization tool referenced throughout this paper as ‘tmvis’, facilitates visualization of these changes by analyzing all mementos in a TimeMap and picking the most unique mementos, which best describe the major changes in a webpage. A web service with a user friendly interface and command line tools are also provided for this tool.

Contents

1. Introduction **4**

2. Design & Implementation **8**

2.1 SimHash 9

2.2 System Architecture 13

2.2.1 TimeMap Fetching 13

2.2.2 SimHash Generation on each memento 16

2.2.3 Unique mementos filtering based on Hamming distance 18

2.2.4 Thumbnail Creation 20

3. Visualizations & System Walkthrough **23**

3.1 Visualizations 23

3.2 System Walkthrough 25

4. Technology Stack & Deployment Details **29**

4.1 Technology Stack 29

4.2 Service Usage 29

5. Future Enhancements **32**

5.1 Downloadable Animated GIF 32

5.2 Embeddable plug and play service 32

5.3 User control over representative thumbnails selection 33

6. Conclusion **33**

References **34**

Appendix A **35**

Appendix B **41**

1. Introduction

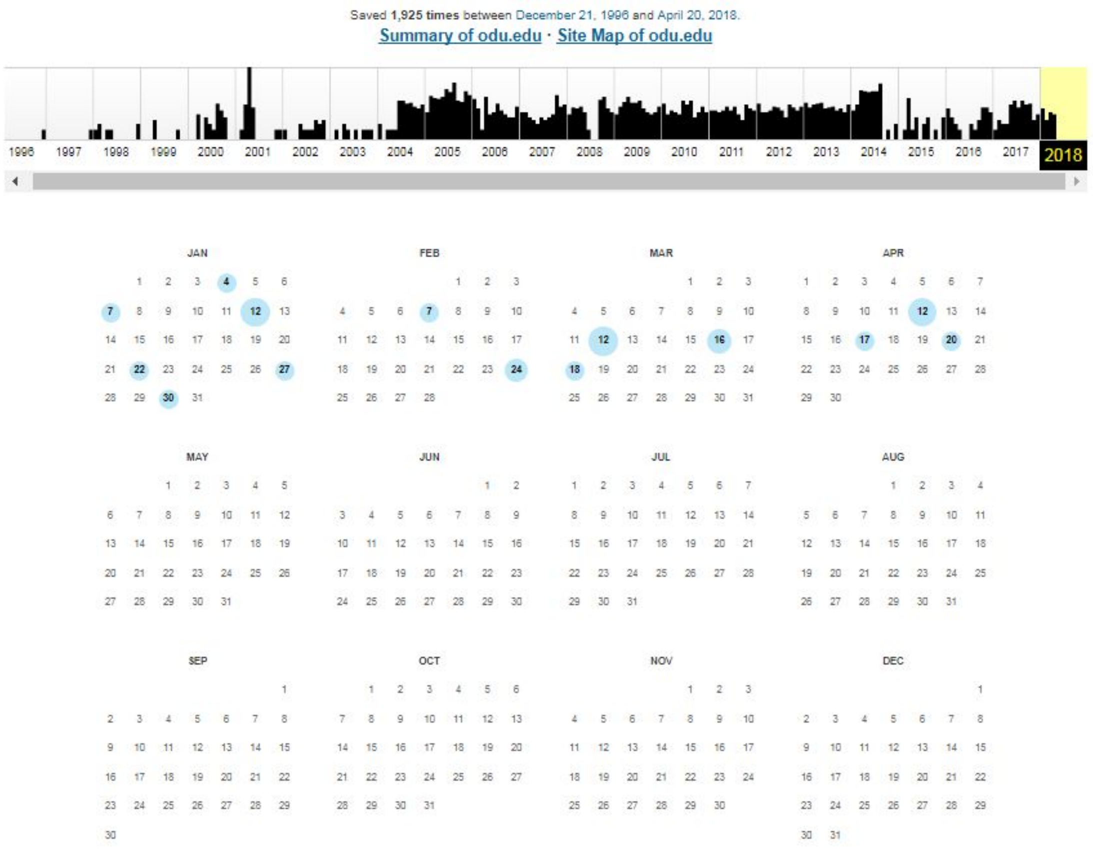
The novel idea behind this project is to address the problem of how to empower the human eye to see the ever changing content of an archived webpage easily.

The web today is dynamic; the content of a webpage today may not be visible tomorrow, basically overwritten by new content. This is because of excessive competition in the field, a website is viewed more when most recent updates are published. It may be in the field of general news, sports, politics, fashion or trends. We have to be thankful for organizations like the Internet Archive[1], Archive-It[2], Archive.is[3], etc. These organizations strive to save as much data on the webpages as possible to help researchers. These archived resources can then be used to retrieve the past content and perceive how a webpage looked like during any preserved time in the past. These organizations act like our Digital Libraries and save our history in a digitized form. This preserved past web becomes helpful for researchers in the field of humanities and social science.

A memento [5] is an archived version of a webpage, preserved at a particular time, or datetime. The TimeMap of a webpage is the list of all the mementos of a particular webpage. A thumbnail is a small image representation of a rendered archived page taken as a snapshot. The three visualization techniques [6] in this project, Grid, Timeline and Slider views, are based on the thumbnail representation of mementos.

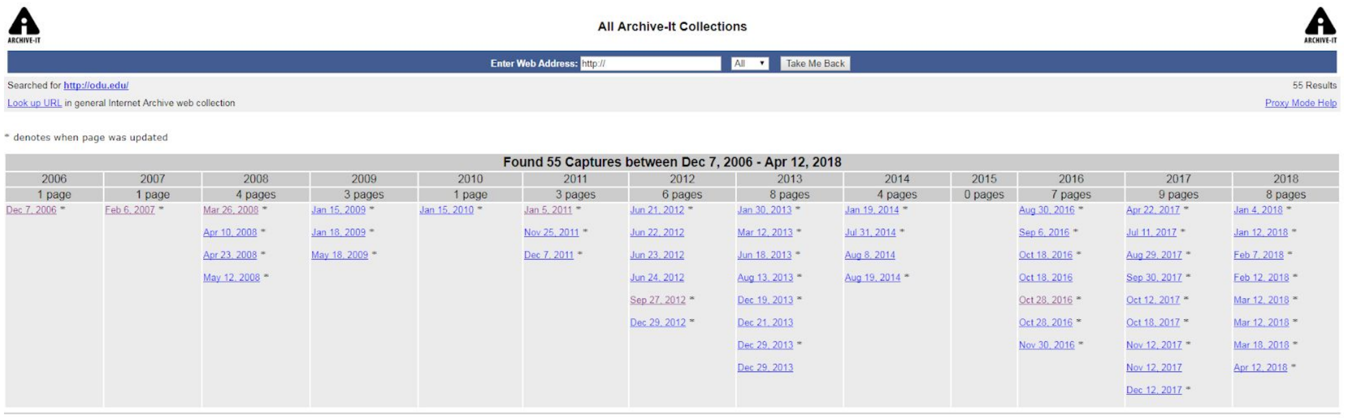
Figures 1 and 2 shows examples of how a user can view a TimeMap, the list of mementos of a particular URI.

[1] http://web.archive.org/
[2] http://archive-it.org/
[3] http://archive.is/


Saved **1,925 times** between December 21, 1996 and April 20, 2018.

Summary of odu.edu · Site Map of odu.edu

Fig. 1: Calendar-style interface from Wayback Machine requested on odu.edu


All Archive-It Collections

Enter Web Address: http:// All Take Me Back

Searched for http://odu.edu/ — 55 Results

Look up URL in general Internet Archive web collection — Proxy Mode Help

* denotes when page was updated

Found 55 Captures between Dec 7, 2006 - Apr 12, 2018

2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
1 page	1 page	4 pages	3 pages	1 page	3 pages	6 pages	8 pages	4 pages	0 pages	7 pages	9 pages	8 pages
Dec 7, 2006 *	Feb 6, 2007 *	Mar 26, 2008 *	Jan 15, 2009 *	Jan 15, 2010 *	Jan 5, 2011 *	Jun 21, 2012 *	Jan 30, 2013 *	Jan 19, 2014 *		Aug 30, 2016 *	Apr 22, 2017 *	Jan 4, 2018 *
		Apr 10, 2008 *	Jan 18, 2009 *		Nov 25, 2011 *	Jun 22, 2012	Mar 12, 2013 *	Jul 31, 2014 *		Sep 6, 2016 *	Jul 11, 2017 *	Jan 12, 2018 *
		Apr 23, 2008 *	May 18, 2009 *		Dec 7, 2011 *	Jun 23, 2012	Jun 18, 2013 *	Aug 8, 2014		Oct 18, 2016 *	Aug 29, 2017 *	Feb 7, 2018 *
		May 12, 2008 *				Jun 24, 2012	Aug 13, 2013 *	Aug 19, 2014 *		Oct 18, 2016	Sep 30, 2017 *	Feb 12, 2018 *
						Sep 27, 2012 *	Dec 19, 2013 *			Oct 28, 2016 *	Oct 12, 2017 *	Mar 12, 2018 *
						Dec 29, 2012 *	Dec 21, 2013			Oct 28, 2016 *	Oct 18, 2017 *	Mar 12, 2018 *
							Dec 29, 2013 *			Nov 30, 2016 *	Nov 12, 2017 *	Mar 18, 2018 *
							Dec 29, 2013				Nov 12, 2017	Apr 12, 2018 *
											Dec 12, 2017 *	

Fig. 2: TimeMap requested from Archive-It on odu.edu

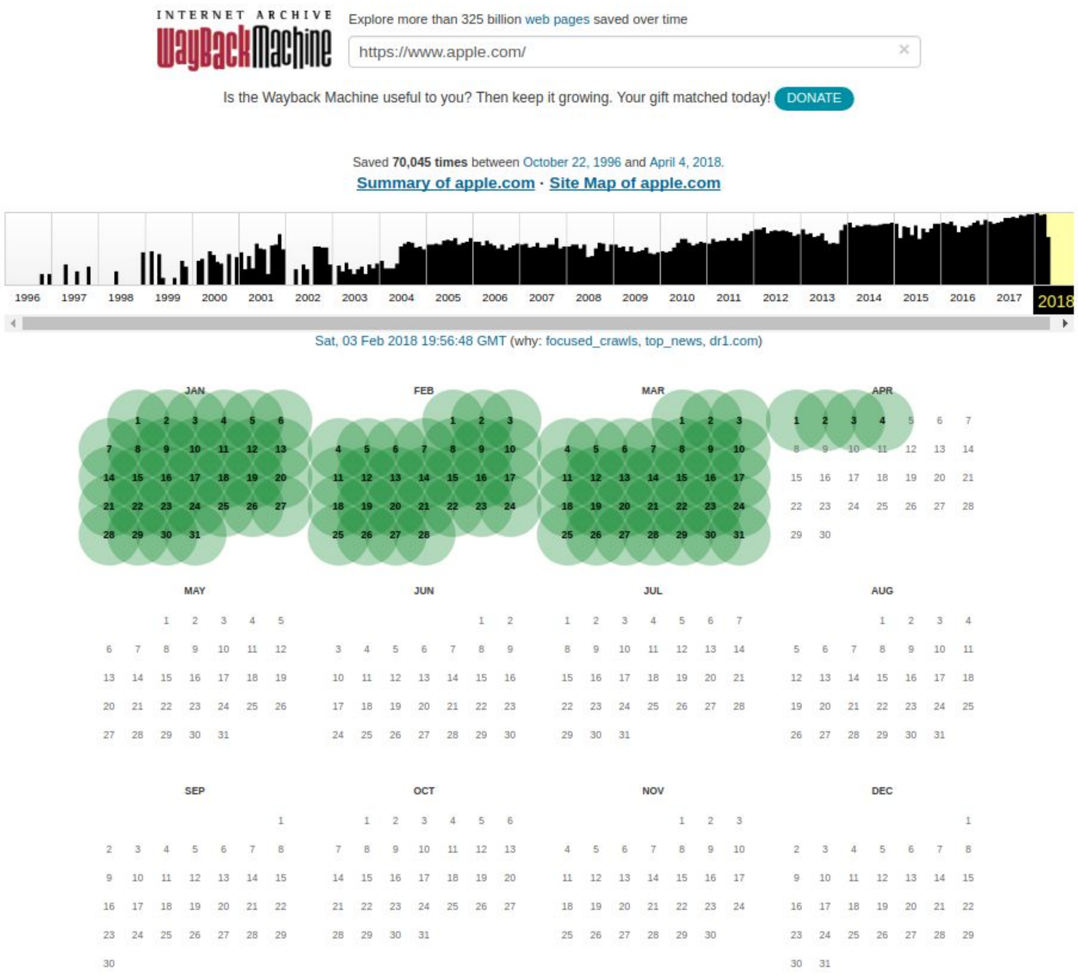
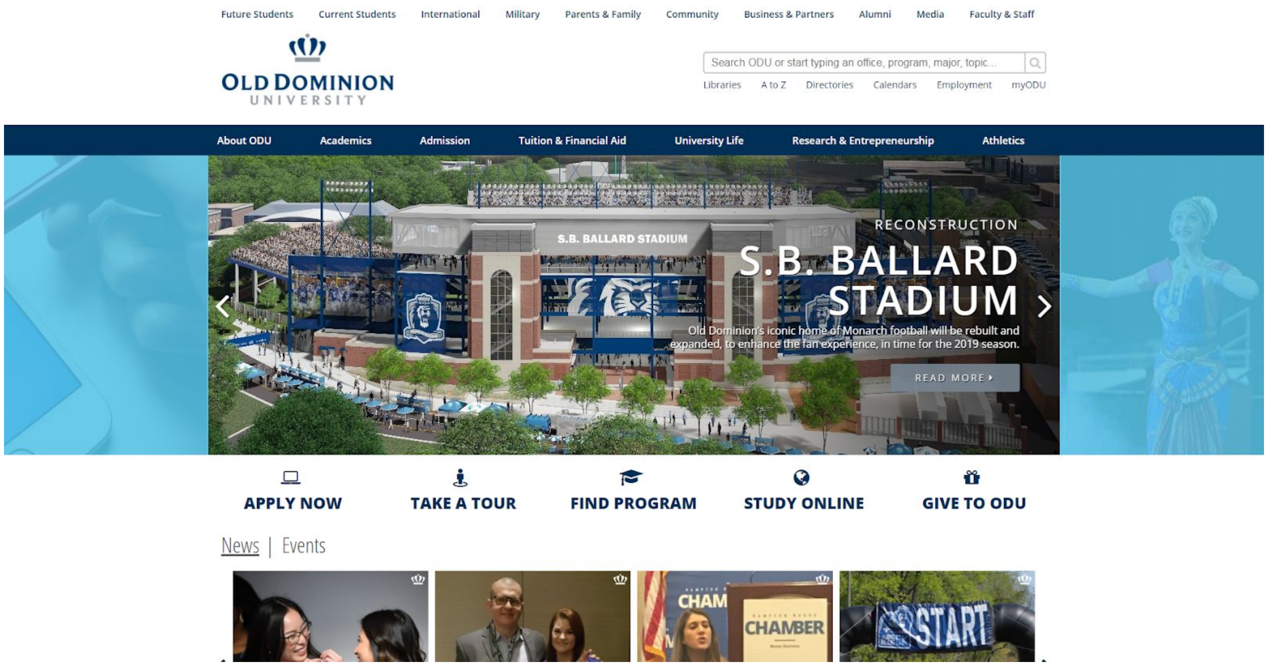

INTERNET ARCHIVE
WaybackMachine
Explore more than 325 billion web pages saved over time
https://www.apple.com/
Is the Wayback Machine useful to you? Then keep it growing. Your gift matched today! DONATE
Saved 70,045 times between October 22, 1996 and April 4, 2018.
Summary of apple.com · Site Map of apple.com
Sat, 03 Feb 2018 19:56:48 GMT (why: focused_crawls, top_news, dr1.com)
2018

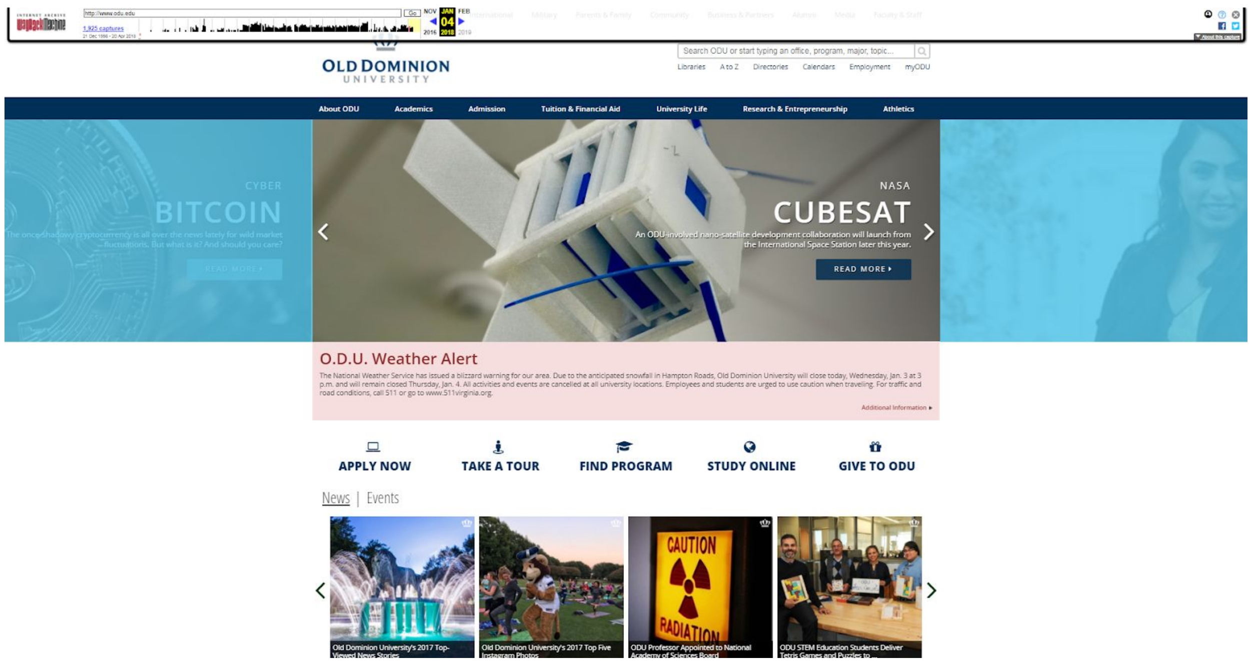
Fig. 3: Calendar-style interface from Wayback Machine requested on apple.com

Only the Internet Archive and Archive-It are presented above although there exist many other archiving organizations. Fig. 1 depicts the mementos on the calendar for the URI odu.edu from the Internet Archive. Fig. 2 depicts the mementos available in collection “all” for the URI odu.edu from Archive-It. It is important to think about the URIs that are popular and that can have several thousand archived copies. Fig. 3 shows the calendar view of the mementos for apple.com. As one can see there are 70,000 mementos. By clicking on any of the bubbles or links, the user would then be navigated to the version of the webpage on that particular datetime and can view exactly how the page looked then. Fig. 4 shows the memento of the page odu.edu from Internet Archive on January 04, 2018, which is obtained by clicking the bubble on Jan 4 in Fig. 1. Fig. 5 shows the live version of same page, odu.edu.



Old Dominion's iconic home of Monarch football will be rebuilt and expanded, to enhance the fan experience, in time for the 2019 season.

Fig. 4: Thumbnail of the memento (January 4, 2018) from Internet Archive of odu.edu



An ODU-involved nano-satellite development collaboration will launch from the International Space Station later this year.

O.D.U. Weather Alert

The National Weather Service has issued a blizzard warning for our area. Due to the anticipated snowfall in Hampton Roads, Old Dominion University will close today, Wednesday, Jan. 3 at 3 p.m. and will remain closed Thursday, Jan. 4. All activities and events are cancelled at all university locations. Employees and students are urged to use caution when traveling. For traffic and road conditions, call 511 or go to www.511virginia.org.

Additional Information

Old Dominion University's 2017 Top-Viewed News Stories

Old Dominion University's 2017 Top Five Instagram Photos

ODU Professor Appointed to National Academy of Sciences Board

ODU STEM Education Students Deliver Tetris Games and Puzzles to ...

Fig. 5: Thumbnail of the live version of the page odu.edu

It is easier to gain an overview of the webpage changes if URI has fewer number of archived versions, like 4 or 5 mementos, because the user can manually render these mementos. This situation quickly worsens if the number of mementos to deal with is large, like when the number of mementos is greater than 50. However, to deal with popular pages like www.apple.com, the task of trying to visualize the summary of what happened is nearly impossible if done manually. The problem this project addresses is to be able to summarize the entire TimeMap and give reliable context to a webpage's evolution. This is useful for a user who wishes to view how a web page changed over time by filtering the unique mementos over the whole TimeMap.

2. Design & Implementation

First, the problem that we are addressing is to be able to select the unique mementos from the TimeMap. The assumption made here is that the underlying HTML content is largely responsible for the way that a web page looks. Considering two mementos, the measure of how different the mementos look can be determined by examining the memento HTML content. If the underlying HTML content differs a lot, the two mementos will look different when rendered. Alsum [1] found that SimHash[4] computed on the HTML content of the memento was effective in measuring the similarities between the webpages.

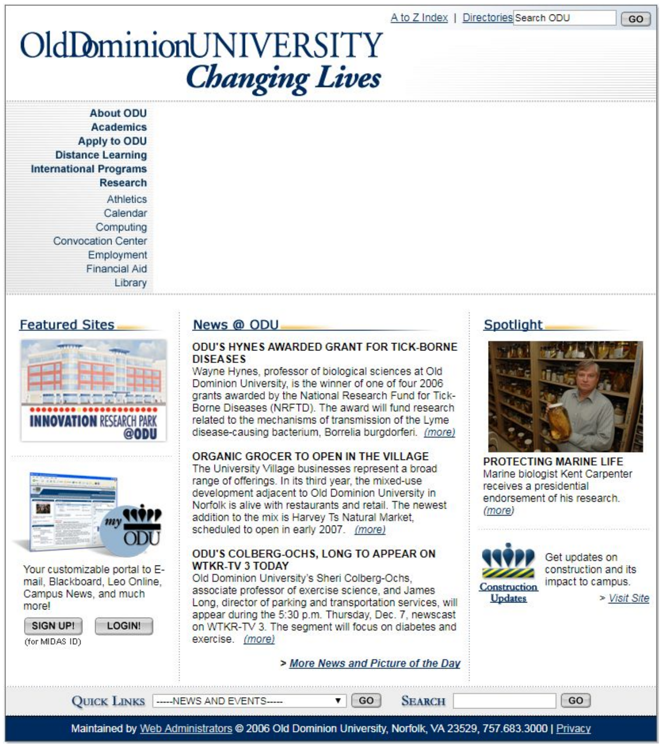
This project is built on top of the code written by Mat Kelly [2], and the visualizations are expanded upon Shankar's Masters Project [3]. Before we delve more into design and implementation aspects, the following section focuses on the SimHash technique, which is a core to this project.

2.1 SimHash

In order to compare the HTML content of two mementos, first the HTML content of the mementos have to be collected. Then a signature fingerprint on whole HTML content is produced through SimHash technique and is compared with the signature of the other memento.

In order to demonstrate why SimHash is a better measure in calculating the webpage similarities than other hash techniques like MD5, three different mementos from the TimeMap of www.odu.edu are requested from the Archive-It, namely memento M1, M2 and M3. The thumbnails are generated and their respective URI-Ms are presented in Fig.7 - 9.

[4] https://en.wikipedia.org/wiki/SimHash


A to Z Index | Directories Search ODU GO

OldDominionUNIVERSITY
Changing Lives

About ODU
Academics
Apply to ODU
Distance Learning
International Programs
Research
Athletics
Calendar
Computing
Convocation Center
Employment
Financial Aid
Library

Featured Sites

INNOVATION RESEARCH PARK @ODU

Your customizable portal to E-mail, Blackboard, Leo Online, Campus News, and much more!

SIGN UP! LOGIN!
(for MIDAS ID)

News @ ODU

ODU'S HYNES AWARDED GRANT FOR TICK-BORNE DISEASES
Wayne Hynes, professor of biological sciences at Old Dominion University, is the winner of one of four 2006 grants awarded by the National Research Fund for Tick-Borne Diseases (NRFTD). The award will fund research related to the mechanisms of transmission of the Lyme disease-causing bacterium, Borrelia burgdorferi. (more)

ORGANIC GROCER TO OPEN IN THE VILLAGE
The University Village businesses represent a broad range of offerings. In its third year, the mixed-use development adjacent to Old Dominion University in Norfolk is alive with restaurants and retail. The newest addition to the mix is Harvey Ts Natural Market, scheduled to open in early 2007. (more)

ODU'S COLBERG-OCHS, LONG TO APPEAR ON WTKR-TV 3 TODAY
Old Dominion University's Sheri Colberg-Ochs, associate professor of exercise science, and James Long, director of parking and transportation services, will appear during the 5:30 p.m. Thursday, Dec. 7, newscast on WTKR-TV 3. The segment will focus on diabetes and exercise. (more)

> More News and Picture of the Day

Spotlight

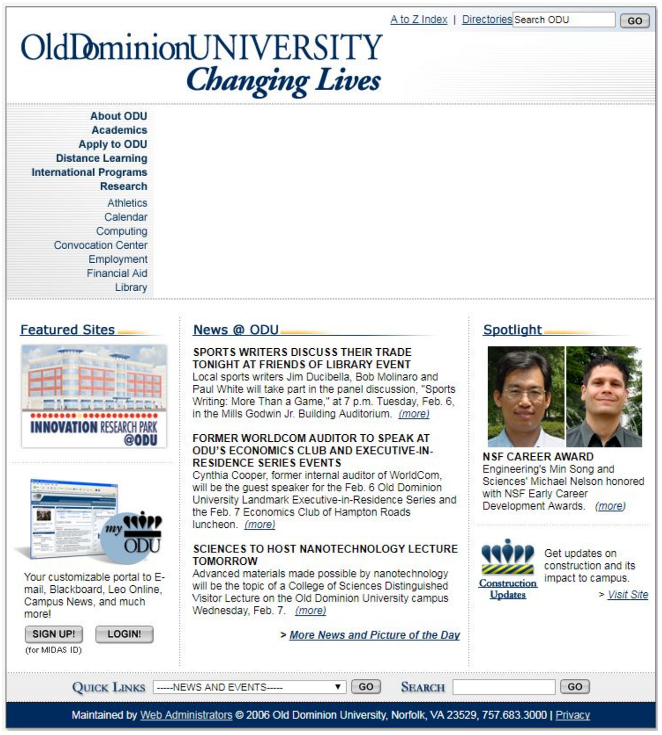
PROTECTING MARINE LIFE
Marine biologist Kent Carpenter receives a presidential endorsement of his research. (more)

Construction Updates
Get updates on construction and its impact to campus.
> Visit Site

QUICK LINKS -----NEWS AND EVENTS----- GO SEARCH GO

Maintained by Web Administrators © 2006 Old Dominion University, Norfolk, VA 23529, 757.683.3000 | Privacy

Fig. 7: Thumbnail of memento M1 (http://wayback.archive-it.org/all/20100115160223if_/http://www.odu.edu/)


A to Z Index | Directories Search ODU GO

OldDominionUNIVERSITY
Changing Lives

About ODU
Academics
Apply to ODU
Distance Learning
International Programs
Research
Athletics
Calendar
Computing
Convocation Center
Employment
Financial Aid
Library

Featured Sites

INNOVATION RESEARCH PARK @ODU

Your customizable portal to E-mail, Blackboard, Leo Online, Campus News, and much more!

SIGN UP! LOGIN!
(for MIDAS ID)

News @ ODU

SPORTS WRITERS DISCUSS THEIR TRADE TONIGHT AT FRIENDS OF LIBRARY EVENT
Local sports writers Jim Ducibella, Bob Molinaro and Paul White will take part in the panel discussion, "Sports Writing: More Than a Game," at 7 p.m. Tuesday, Feb. 6, in the Mills Godwin Jr. Building Auditorium. (more)

FORMER WORLDCOM AUDITOR TO SPEAK AT ODU'S ECONOMICS CLUB AND EXECUTIVE-IN-RESIDENCE SERIES EVENTS
Cynthia Cooper, former internal auditor of WorldCom, will be the guest speaker for the Feb. 6 Old Dominion University Landmark Executive-in-Residence Series and the Feb. 7 Economics Club of Hampton Roads luncheon. (more)

SCIENCES TO HOST NANOTECHNOLOGY LECTURE TOMORROW
Advanced materials made possible by nanotechnology will be the topic of a College of Sciences Distinguished Visitor Lecture on the Old Dominion University campus Wednesday, Feb. 7. (more)

> More News and Picture of the Day

Spotlight

NSF CAREER AWARD
Engineering's Min Song and Sciences' Michael Nelson honored with NSF Early Career Development Awards. (more)

Construction Updates
Get updates on construction and its impact to campus.
> Visit Site

QUICK LINKS -----NEWS AND EVENTS----- GO SEARCH GO

Maintained by Web Administrators © 2006 Old Dominion University, Norfolk, VA 23529, 757.683.3000 | Privacy

Fig. 8: Thumbnail of memento M2 (http://wayback.archive-it.org/all/20110105165700if_/http://www.odu.edu/)

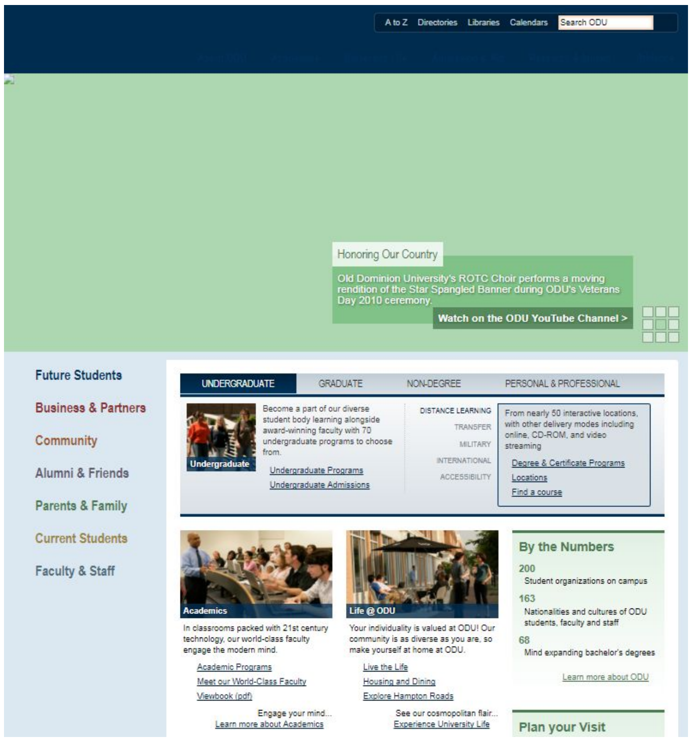


Fig. 9: Thumbnail of memento M3 (http://wayback.archive-it.org/all/20080326001151if_/http://www.odu.edu/)

By looking at these thumbnails, one can say that the thumbnails for memento M1 and M2 look alike, whereas M3 is far distinct from the other two. At a closer look, one will realize that M1 and M2 differ with the content they hold, though the layout is the same. Now an attempt is made to compute the hashes on the HTML content of these selected mementos. At first MD5 hashing is used to compute the hashes and is as shown below:

```
$curl URI-M1 | md5sum  ->  fc8e53aebb9061f390aba82665581295
$curl URI-M2 | md5sum  ->  d546e192eab633f4d1b4451399c8adcc
$curl URI-M3 | md5sum  ->  5e98bc5367c86f3ffaea0b8c3deb3f5d
```

After the hashes are generated, the Hamming distance is calculated between the pairs (M1, M2) and (M2, M3). The Hamming distance is the minimum number of substitutions required to convert one string to another. The results are as shown below:

```
$ node hammingdistance M1 M2 ->  30
$ node hammingdistance M2 M3 ->  32
```

Now the same process is repeated with SimHash being the hashing technique used and the results are as shown below:

```
$curl URI-M1 | SimHash  ->  8c27981eaed151cfa645ad823932eac6
$curl URI-M2 | SimHash  ->  8c27981faad951cf8645ad823d32eac2
$curl URI-M3 | SimHash  ->  fa3799170258494b9443b9be3977a84e
```

```
$ node hammingdistance M1 M2 ->  6
$ node hammingdistance M2 M3 -> 27
```

On a closer look at the Hamming distance outputs, notice that the SimHashes follows a notion of Hamming distance being small (6) when the difference between the mementos content (**M1**, **M2**) is small, and the Hamming distance gets larger (27) when the difference between the mementos (**M2**, **M3**) is larger.

This notion does not hold with the MD5 hashes. Even though the difference between HTML contents of the memento **M1** and **M2** is small, the Hamming distance between MD5 hashes of **M1, M2** is much larger. Hence we can conclude that SimHash correlates better with the webpage similarity.

2.2 System Architecture

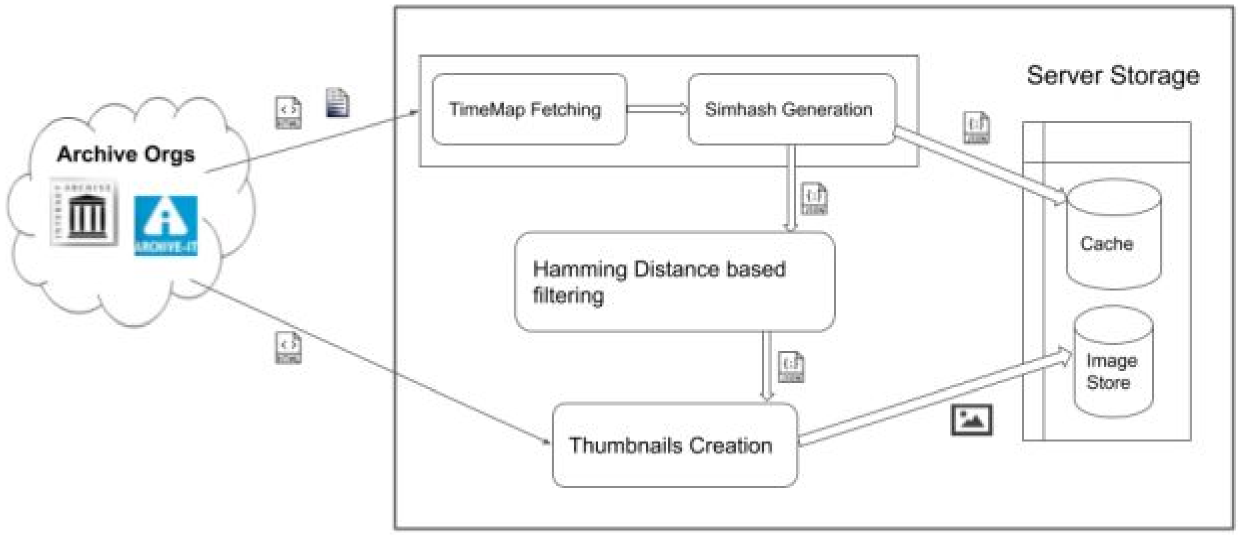


Fig. 10: System Architecture of tmvis

The TimeMap summarization tool developed is referenced as ‘tmvis’ throughout this paper. The system at the server side has four stages. As shown in Fig. 10, the Internet Archive and Archive-It are the prime sources used in this system for the archived content. Each of these stages are explained in the following sections in detail. Server Storage is used as a cache to reduce the time to serve the results when the same URI is requested for the summarization in future. All the thumbnails created are stored in the Image Store, and SimHashes generated are stored in the Cache, so that it does not have to be computed again. Below, we present the details of each of these stages.

2.2.1 TimeMap Fetching

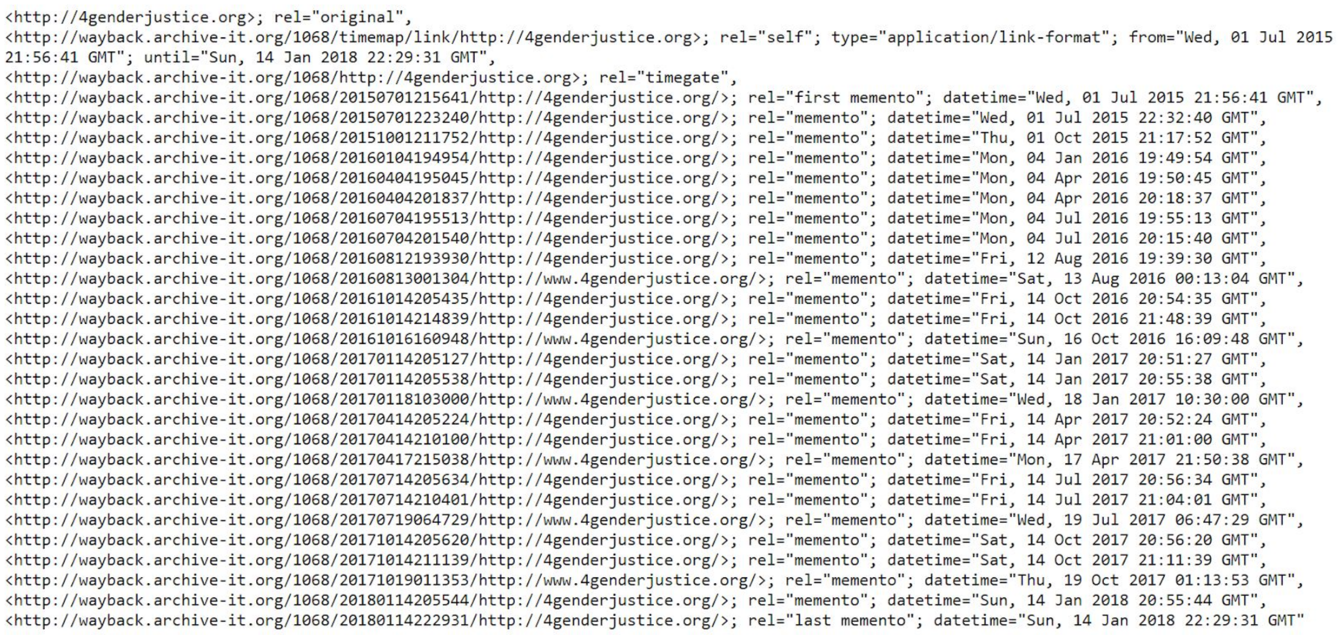
The backend for this project is built on top of the base code done by Mat Kelly [2], which is written in Node.js. In this project, both Internet Archive and Archive-It are considered to be the prime sources for fetching the archived content of a user provided URI-R. First, when an URI is given as an input to the UI, the prime source is chosen as either Internet Archive or Archive-It. If Archive-It is chosen as a primary source an additional attribute, collection number can be specified; if nothing is specified, the argument “all” is considered to be the default value.

Once the request is made for summarization, the user inputs, such as URI-R, opted source, and collection number, are gathered. An HTTP request is made to the opted source (either Internet Archive or Archive-It) to fetch the whole TimeMap of the requested URI-R. For example, for the URI-R http://4genderjustice.org, URIs to fetch the TimeMap are as follows:

1. wayback.archive-it.org/1068/timemap/link/http://4genderjustice.org
2. wayback.archive-it.org/all/timemap/link/http://4genderjustice.org
3. web.archive.org/web/timemap/link/https://4genderjustice.org

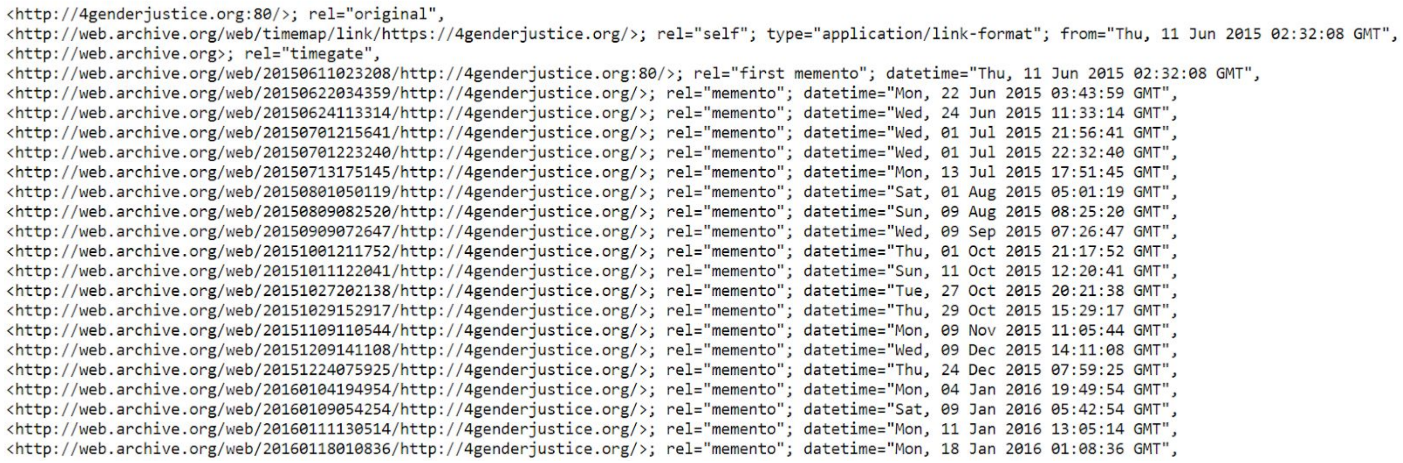
URI 1 is to fetch the TimeMap of URI-R http://4genderjustice.org from Archive-It from the collection number 1068. URI 2 is to fetch the TimeMap of URI-R http://4genderjustice.org from Archive-It, and collection number ‘all’ is the default used in tmvis when the nothing is provided by the user. The 3rd URI is the request for the TimeMap from the Internet Archive.

The response for the HTTP request is the whole TimeMap. Fig. 11 and 12 are partial snapshots of TimeMaps, one each from Internet Archive and Archive-It.



```
<http://4genderjustice.org>; rel="original",
<http://wayback.archive-it.org/1068/timemap/link/http://4genderjustice.org>; rel="self"; type="application/link-format"; from="Wed, 01 Jul 2015
21:56:41 GMT"; until="Sun, 14 Jan 2018 22:29:31 GMT",
<http://wayback.archive-it.org/1068/http://4genderjustice.org>; rel="timegate",
<http://wayback.archive-it.org/1068/20150701215641/http://4genderjustice.org/>; rel="first memento"; datetime="Wed, 01 Jul 2015 21:56:41 GMT",
<http://wayback.archive-it.org/1068/20150701223240/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 01 Jul 2015 22:32:40 GMT",
<http://wayback.archive-it.org/1068/20151001211752/http://4genderjustice.org/>; rel="memento"; datetime="Thu, 01 Oct 2015 21:17:52 GMT",
<http://wayback.archive-it.org/1068/20160104194954/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Jan 2016 19:49:54 GMT",
<http://wayback.archive-it.org/1068/20160404195045/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Apr 2016 19:50:45 GMT",
<http://wayback.archive-it.org/1068/20160404201837/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Apr 2016 20:18:37 GMT",
<http://wayback.archive-it.org/1068/20160704195513/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Jul 2016 19:55:13 GMT",
<http://wayback.archive-it.org/1068/20160704201540/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Jul 2016 20:15:40 GMT",
<http://wayback.archive-it.org/1068/20160812193930/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 12 Aug 2016 19:39:30 GMT",
<http://wayback.archive-it.org/1068/20160813001304/http://www.4genderjustice.org/>; rel="memento"; datetime="Sat, 13 Aug 2016 00:13:04 GMT",
<http://wayback.archive-it.org/1068/20161014205435/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Oct 2016 20:54:35 GMT",
<http://wayback.archive-it.org/1068/20161014214839/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Oct 2016 21:48:39 GMT",
<http://wayback.archive-it.org/1068/20161016160948/http://www.4genderjustice.org/>; rel="memento"; datetime="Sun, 16 Oct 2016 16:09:48 GMT",
<http://wayback.archive-it.org/1068/20170114205127/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 14 Jan 2017 20:51:27 GMT",
<http://wayback.archive-it.org/1068/20170114205538/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 14 Jan 2017 20:55:38 GMT",
<http://wayback.archive-it.org/1068/20170118103000/http://www.4genderjustice.org/>; rel="memento"; datetime="Wed, 18 Jan 2017 10:30:00 GMT",
<http://wayback.archive-it.org/1068/20170414205224/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Apr 2017 20:52:24 GMT",
<http://wayback.archive-it.org/1068/20170414210100/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Apr 2017 21:01:00 GMT",
<http://wayback.archive-it.org/1068/20170417215038/http://www.4genderjustice.org/>; rel="memento"; datetime="Mon, 17 Apr 2017 21:50:38 GMT",
<http://wayback.archive-it.org/1068/20170714205634/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Jul 2017 20:56:34 GMT",
<http://wayback.archive-it.org/1068/20170714210401/http://4genderjustice.org/>; rel="memento"; datetime="Fri, 14 Jul 2017 21:04:01 GMT",
<http://wayback.archive-it.org/1068/20170719064729/http://www.4genderjustice.org/>; rel="memento"; datetime="Wed, 19 Jul 2017 06:47:29 GMT",
<http://wayback.archive-it.org/1068/20171014205620/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 14 Oct 2017 20:56:20 GMT",
<http://wayback.archive-it.org/1068/20171014211139/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 14 Oct 2017 21:11:39 GMT",
<http://wayback.archive-it.org/1068/20171019011353/http://www.4genderjustice.org/>; rel="memento"; datetime="Thu, 19 Oct 2017 01:13:53 GMT",
<http://wayback.archive-it.org/1068/20180114205544/http://4genderjustice.org/>; rel="memento"; datetime="Sun, 14 Jan 2018 20:55:44 GMT",
<http://wayback.archive-it.org/1068/20180114222931/http://4genderjustice.org/>; rel="last memento"; datetime="Sun, 14 Jan 2018 22:29:31 GMT"
```

Fig. 11: TimeMap snapshot for 4genderjustice.org from Archive-It



```
<http://4genderjustice.org:80/>; rel="original",
<http://web.archive.org/web/timemap/link/https://4genderjustice.org/>; rel="self"; type="application/link-format"; from="Thu, 11 Jun 2015 02:32:08 GMT",
<http://web.archive.org>; rel="timegate",
<http://web.archive.org/web/20150611023208/http://4genderjustice.org:80/>; rel="first memento"; datetime="Thu, 11 Jun 2015 02:32:08 GMT",
<http://web.archive.org/web/20150622034359/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 22 Jun 2015 03:43:59 GMT",
<http://web.archive.org/web/20150624113314/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 24 Jun 2015 11:33:14 GMT",
<http://web.archive.org/web/20150701215641/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 01 Jul 2015 21:56:41 GMT",
<http://web.archive.org/web/20150701223240/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 01 Jul 2015 22:32:40 GMT",
<http://web.archive.org/web/20150713175145/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 13 Jul 2015 17:51:45 GMT",
<http://web.archive.org/web/20150801050119/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 01 Aug 2015 05:01:19 GMT",
<http://web.archive.org/web/20150809082520/http://4genderjustice.org/>; rel="memento"; datetime="Sun, 09 Aug 2015 08:25:20 GMT",
<http://web.archive.org/web/20150909072647/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 09 Sep 2015 07:26:47 GMT",
<http://web.archive.org/web/20151001211752/http://4genderjustice.org/>; rel="memento"; datetime="Thu, 01 Oct 2015 21:17:52 GMT",
<http://web.archive.org/web/20151011122041/http://4genderjustice.org/>; rel="memento"; datetime="Sun, 11 Oct 2015 12:20:41 GMT",
<http://web.archive.org/web/20151027202138/http://4genderjustice.org/>; rel="memento"; datetime="Tue, 27 Oct 2015 20:21:38 GMT",
<http://web.archive.org/web/20151029152917/http://4genderjustice.org/>; rel="memento"; datetime="Thu, 29 Oct 2015 15:29:17 GMT",
<http://web.archive.org/web/20151109110544/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 09 Nov 2015 11:05:44 GMT",
<http://web.archive.org/web/20151209141108/http://4genderjustice.org/>; rel="memento"; datetime="Wed, 09 Dec 2015 14:11:08 GMT",
<http://web.archive.org/web/20151224075925/http://4genderjustice.org/>; rel="memento"; datetime="Thu, 24 Dec 2015 07:59:25 GMT",
<http://web.archive.org/web/20160104194954/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 04 Jan 2016 19:49:54 GMT",
<http://web.archive.org/web/20160109054254/http://4genderjustice.org/>; rel="memento"; datetime="Sat, 09 Jan 2016 05:42:54 GMT",
<http://web.archive.org/web/20160111130514/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 11 Jan 2016 13:05:14 GMT",
<http://web.archive.org/web/20160118010836/http://4genderjustice.org/>; rel="memento"; datetime="Mon, 18 Jan 2016 01:08:36 GMT",
```

Fig. 12: TimeMap snapshot for 4genderjustice.org from Internet Archive

As shown in Fig. 11, the TimeMap consists of list of all mementos separated by commas, each consisting a URI-M 'rel' and 'datetime' attributes. The first and last memento can be identified by looking at the 'rel' attribute. Along with the list of mementos, the TimeMap also consists of the original URI, self-reference to TimeMap and a TimeGate as well. These different resources are identified by the 'rel' attribute, which is given one of the enumerated values: ‘original’, ‘self’, ‘TimeGate, ‘first memento’, ‘memento’, or ‘last memento’.

Once the TimeMap is obtained, it is then parsed to compile a list of mementos. Each of the mementos from the list are then fetched, and then are sent through a mechanism to compute the SimHash, which is explained in the next section.

2.2.2 SimHash Generation on each memento

In this step, first an HTTP request is made to get the HTML content of each of memento. Only the mementos that return a HTTP 200 status code are considered for further steps in the process.

It is important to know about the options ‘id_’, ‘if_’ that can be appended to the 14 digit datetime part of the URI-M:

- Option ‘id_’ is to get the original HTML content without any URL rewriting or banner insertion.
- Option ‘if_’ is to get the HTML content without the banner, but with links rewritten to point to the archive.

When computing the SimHash on each of the memento content, ‘id_’ is considered while fetching the HTML content of the memento, because our requirement is focused on the original content. That is when SimHash signature generated would actually represent the original archived resource. Example for such URI-M that uses ‘id_’ http://wayback.archive-it.org/all/20100115160223id_/http://www.odu.edu/.

If the HTTP request on such URI-M yields a successful HTTP 200 status code, the whole HTML content is read into a buffer and the SimHash method is called upon this content. The SimHash library[5] that is used in tmvis is from npmjs packages, the version 0.1.0.

To improve the performance of computing the SimHashes, instead of making the HTTP request to obtain the HTML content on one memento at a time, multiple requests can be made in parallel. The selection of Node.js for the backend supports this functionality. At this time the Internet Archive has a restriction of at most 10 requests at a time from a single IP address.

In the special case that HTML content is returned in gzip format, the 'content-encoding' of the response header is checked. The 'zlib' library is used to unzip the content to handle such unusual cases.

Once the 128-bit binary SimHash is computed, it is then converted to a 32 character hexadecimal code and saved in a file that acts as a cache for future requests made on the same URI. In that way the HTML content for all these mementos do not need to be fetched repeatedly. When the server is run using the --oes (Override existing content mode) option, the latest TimeMap is fetched and SimHashes are re-generated for all the mementos. The next step in the process is to compute the Hamming distances and using it to filter the unique mementos. The next section explains this in its greater depth.

For popular webpages, like cnn.com or apple.com, the number of mementos these URIs have are so large that fetching the TimeMap itself takes a quite a bit of time. As an example, cnn.com has more than 180,000 mementos. Downloading the HTML content of each memento in such huge TimeMap takes hours, and it is practically impossible to keep the user waiting for such long time. Hence, the decision is made for this project that only

[5] https://www.npmjs.com/package/SimHash

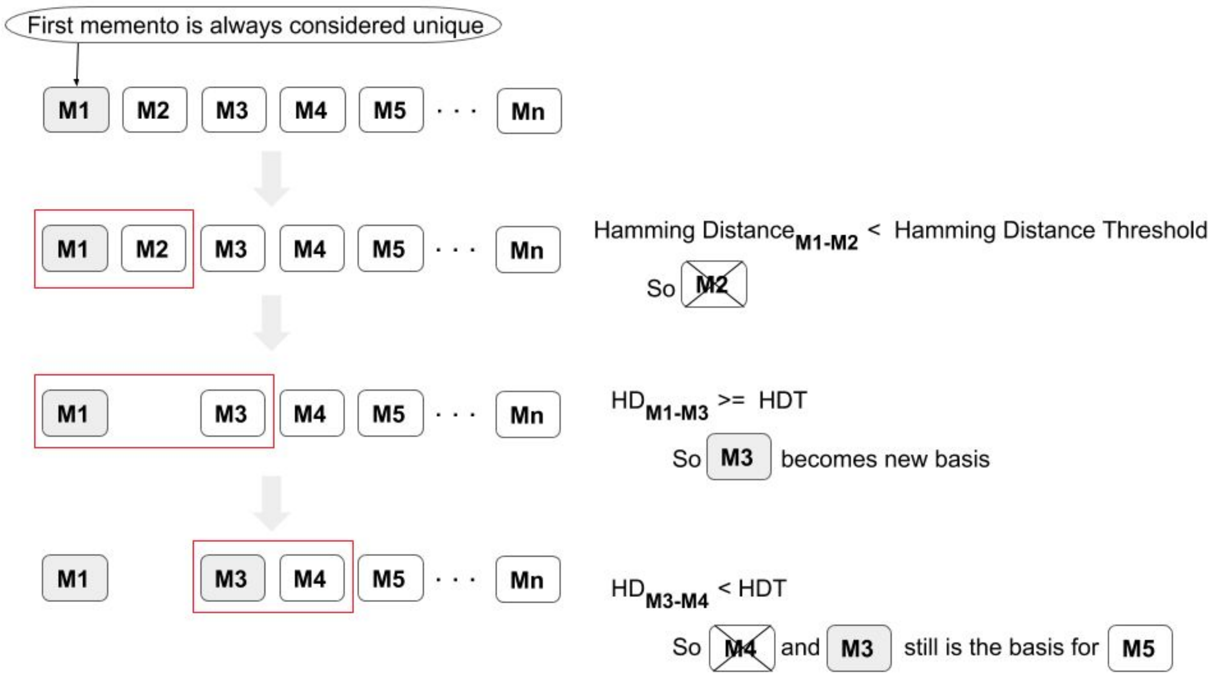
the last 5000 successfully returned mementos are taken into account for the processing and the user is notified of the datetime range for the segment used.

2.2.3 Unique mementos filtering based on Hamming distance

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In other words, it measures the minimum number of substitutions required to change one string into the other.

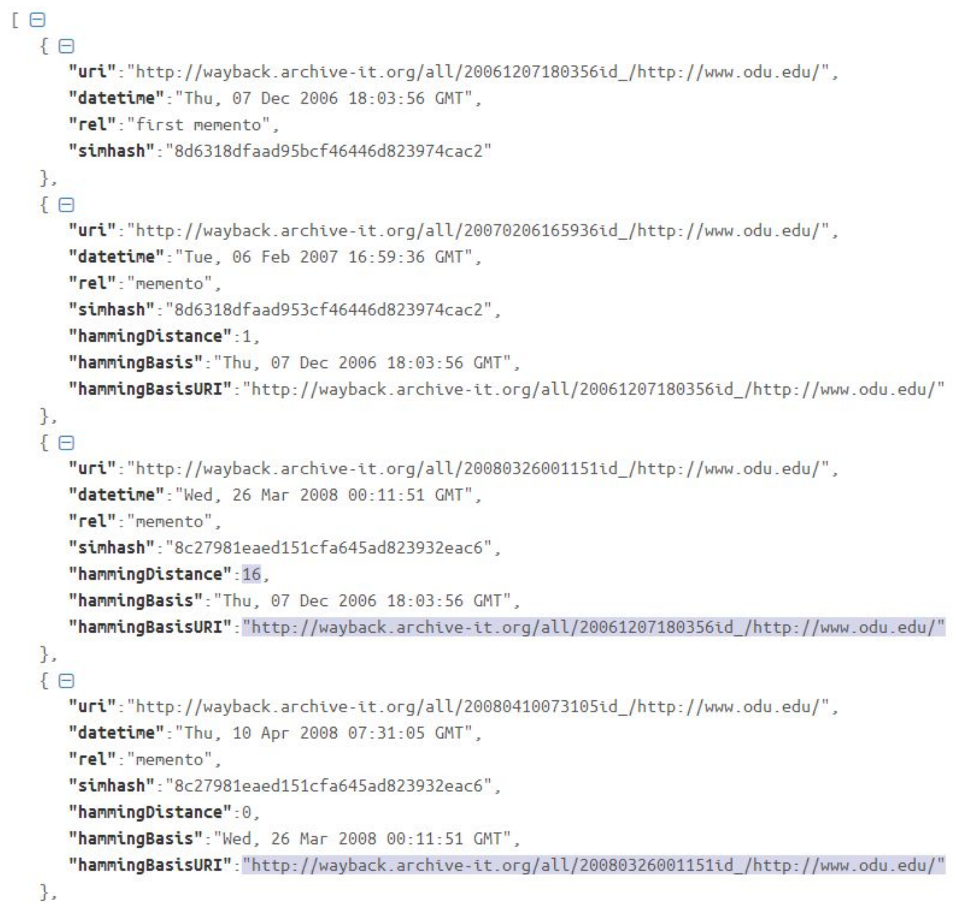
The next step after the SimHashes for all the mementos are obtained is to filter amongst them using the Hamming distance. The filtering process starts by declaring the first memento of the TimeMap to be included in the summarization. This memento acts as the base to compare with the subsequent mementos. The other attribute that is required is to decide upon the Hamming distance threshold. The greater the Hamming distance threshold selected, the more difference between the HTML content of the mementos, which means that the resulting filtered archived webpages are more different from each other.

Consider a scenario of TimeMap of an URI-R that has *n* mementos denoted by **M1**, **M2,** … **Mn**. Let the Hamming distance Threshold be **HDT**. Fig. 13 walks through an example of the selection algorithm. The filtering process starts by considering the first memento of the TimeMap to be unique and considered a representative memento. **M1** becomes the basis to compare with **M2**. If the calculated Hamming distance between **M1** and **M2** is less than **HDT, M2** is ignored and **M1** remains the basis. The Hamming distance between the SimHashes of **M1** and **M3** is calculated, and if this is greater than or equal to HDT, **M3** is considered as a representative memento and becomes the new basis. If the calculated Hamming distance between **M3** and **M4** is calculated and is compared with **HDT**, **M4** is now eliminated as the Hamming distance measured is considered less than **HDT. M3** still is the basis for the next memento which is **M5**. This process continues up to the last memento of the TimeMap. By the end of this filtering process, a list of all mementos with the Hamming distance and the Hamming basis URI for each of the memento is obtained as shown in Fig. 14.



First memento is always considered unique
M1 M2 M3 M4 M5 . . . Mn
Hamming Distance M1-M2 < Hamming Distance Threshold
So M2
HD M1-M3 >= HDT
So M3 becomes new basis
HD M3-M4 < HDT
So M4 and M3 still is the basis for M5

Fig. 13: Selection Algorithm based on Hamming distance Threshold



```
[ ⊟
  { ⊟
    "uri":"http://wayback.archive-it.org/all/20061207180356id_/http://www.odu.edu/",
    "datetime":"Thu, 07 Dec 2006 18:03:56 GMT",
    "rel":"first memento",
    "simhash":"8d6318dfaad95bcf46446d823974cac2"
  },
  { ⊟
    "uri":"http://wayback.archive-it.org/all/20070206165936id_/http://www.odu.edu/",
    "datetime":"Tue, 06 Feb 2007 16:59:36 GMT",
    "rel":"memento",
    "simhash":"8d6318dfaad953cf46446d823974cac2",
    "hammingDistance":1,
    "hammingBasis":"Thu, 07 Dec 2006 18:03:56 GMT",
    "hammingBasisURI":"http://wayback.archive-it.org/all/20061207180356id_/http://www.odu.edu/"
  },
  { ⊟
    "uri":"http://wayback.archive-it.org/all/20080326001151id_/http://www.odu.edu/",
    "datetime":"Wed, 26 Mar 2008 00:11:51 GMT",
    "rel":"memento",
    "simhash":"8c27981eaed151cfa645ad823932eac6",
    "hammingDistance":16,
    "hammingBasis":"Thu, 07 Dec 2006 18:03:56 GMT",
    "hammingBasisURI":"http://wayback.archive-it.org/all/20061207180356id_/http://www.odu.edu/"
  },
  { ⊟
    "uri":"http://wayback.archive-it.org/all/20080410073105id_/http://www.odu.edu/",
    "datetime":"Thu, 10 Apr 2008 07:31:05 GMT",
    "rel":"memento",
    "simhash":"8c27981eaed151cfa645ad823932eac6",
    "hammingDistance":0,
    "hammingBasis":"Wed, 26 Mar 2008 00:11:51 GMT",
    "hammingBasisURI":"http://wayback.archive-it.org/all/20080326001151id_/http://www.odu.edu/"
  },
```

Fig. 14: Part of mementos from www.odu.edu from Archive-It with HDT = 5

In Fig. 14 notice how the Hamming basis for memento **M4** has changed. The reason for this is that the Hamming distance between memento **M2** to memento **M3** is calculated as 16, which is greater than this HDT considered (5), hence memento **M3** becomes a representative memento and becomes the basis to compare with memento **M4**.

2.2.4 Thumbnail Creation

The next step is to create thumbnails of the identified representative mementos. This section explains how these unique representative mementos undergo the process of screenshot capture of their rendered webpage.

As discussed, a thumbnail is a small image representation of an archived page taken as a snapshot. Now the task at hand is to fetch all the mementos marked unique and capture a screenshot of the rendered mementos.

Originally PhantomJS[6] was used for headless browsing. But this library is no longer supported, so we have replaced it with Puppeteer[7]. The way Puppeteer works is that, first a headless browser is launched and a new browser page is instantiated and is passed the URI as the argument. Behind the scenes, an actual browser tab is opened and the URI is rendered. A screenshot request is made on this rendered page. All of this is happening in headless mode, meaning that the actual UI is being simulated. Instead of a user acting on the browser, code is made to interact with these headless browsers. These headless browsers automate the browsing interactions like the opening of a browser, loading the webpage, and capturing the screenshot.

Puppeteer comes with different options on how long to wait before the DOM content is fully loaded. A few of those options for waitUntil are as follows:

- domcontentloaded - wait until the entire DOM content is loaded
- networkidle0 - wait until 0 active flight requests
- networkidle2 - wait until there exist only 2 active flight requests

Different archived webpages have different load times depending on the embedded resources. Even after the entire DOM content is loaded, there is a chance that a few images that the webpage is rendering might take a bit of time. To accommodate such extra time, a delay of 2000ms is provided before Puppeteer actually captures a screenshot. Once a screenshot of the memento is generated, it is preserved and served for all future requests that need the generation of a screenshot for the same memento.

[6] PhantomJS: http://phantomjs.org/documentation/
[7] Puppeteer: https://developers.google.com/web/tools/puppeteer/

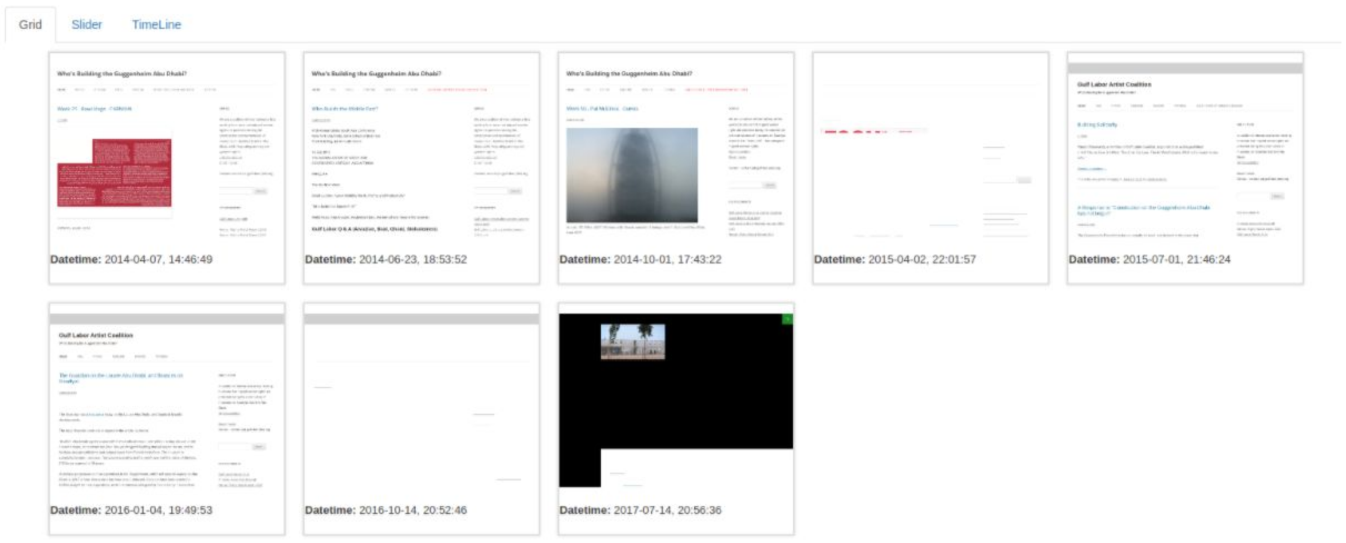

Grid
Slider
TimeLine
Datetime: 2014-04-07, 14:46:49
Datetime: 2014-06-23, 18:53:52
Datetime: 2014-10-01, 17:43:22
Datetime: 2015-04-02, 22:01:57
Datetime: 2015-07-01, 21:46:24
Datetime: 2016-01-04, 19:49:53
Datetime: 2016-10-14, 20:52:46
Datetime: 2017-07-14, 20:56:36

Fig. 15: Summarized http://gulflabor.org/ from tmvis, Archive-It 1068, 0 second delay

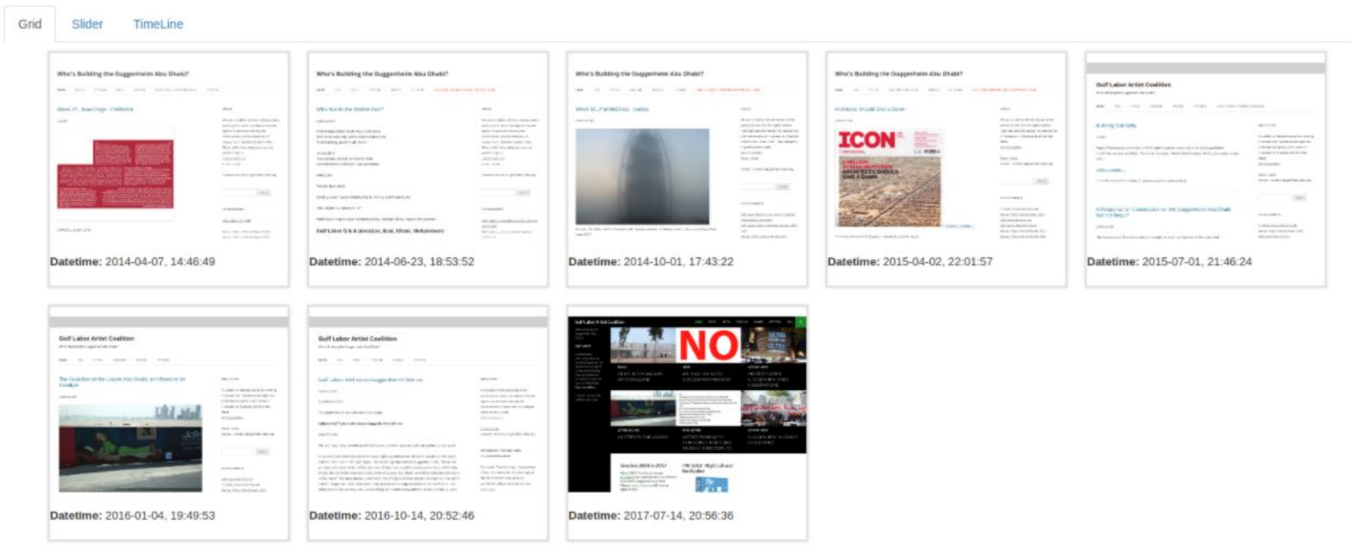

Grid
Slider
TimeLine
Datetime: 2014-04-07, 14:46:49
Datetime: 2014-06-23, 18:53:52
Datetime: 2014-10-01, 17:43:22
Datetime: 2015-04-02, 22:01:57
Datetime: 2015-07-01, 21:46:24
Datetime: 2016-01-04, 19:49:53
Datetime: 2016-10-14, 20:52:46
Datetime: 2017-07-14, 20:56:36

Fig. 16: Summarized http://gulflabor.org/ from tmvis, Archive-It 1068, 2 second delay

Fig. 15 and Fig. 16 show the Image Grid view of tmvis summarizing the TimeMap of http://gulflabor.org/ from collection 1068 of Archive-It. Notice that the thumbnails in Fig. 16 are more fully loaded than in Fig. 15. The only difference made with Fig. 16 was that Puppeteer is made to wait for 2 seconds after it renders the webpage of the memento using the method *page.waitFor*, before the screenshot capture command is issued. Having this delay incorporated, the thumbnails better represent the actual rendered mementos and hence user is provided with clear view of rendered webpage of representative mementos.

3. Visualizations & System Walkthrough

3.1 Visualizations

The UI of this project consists of three different visualizations: Image Grid, Slider, and Timeline. The base code used here is from Shankar [3].

The **Image Grid** is a simple view, where the thumbnails of all the unique mementos are arranged in a left to right, top to bottom manner. The main intention is to show the user all these unique archived versions of the webpage so that the user gets an overview of how the webpage has changed over time. Fig. 17 shows an example of the grid view for www.odu.edu.

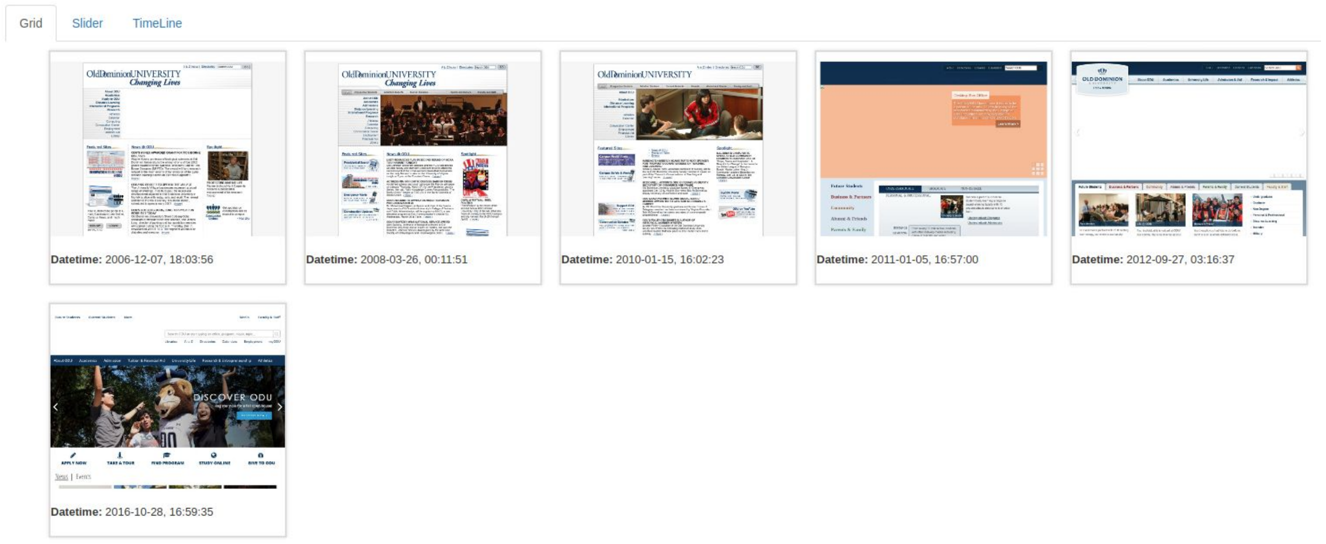
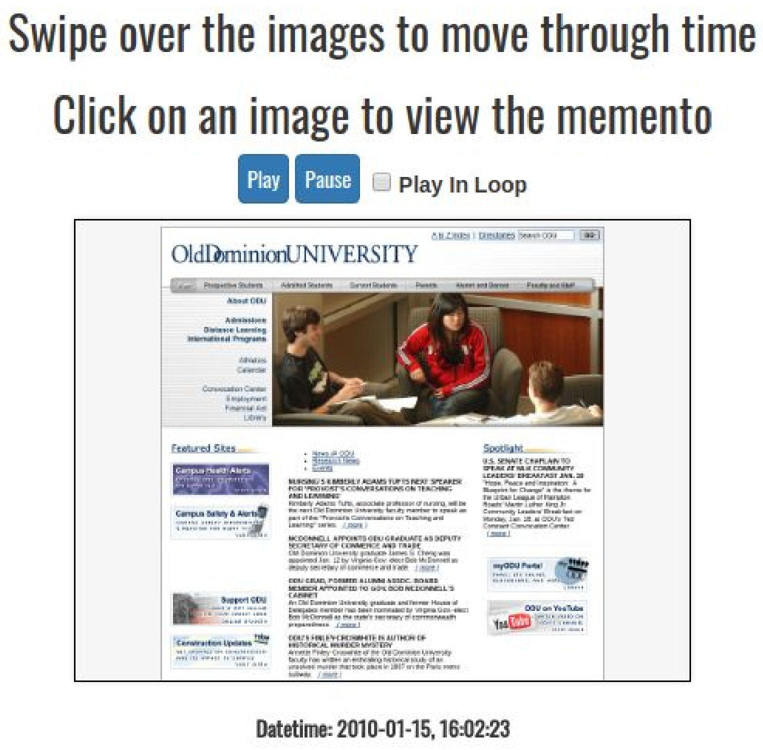

Grid
Slider
TimeLine
Datetime: 2006-12-07, 18:03:56
Datetime: 2008-03-26, 00:11:51
Datetime: 2010-01-15, 16:02:23
Datetime: 2011-01-05, 16:57:00
Datetime: 2012-09-27, 03:16:37
Datetime: 2016-10-28, 16:59:35

Fig. 17: Grid view of http://www.odu.edu generated by tmvis.

The **Image Slider** imitates the photo roller functionality used in iPhoto. By simply moving the cursor across the thumbnail, a different thumbnail in the order is shown to the user. Clicking on the thumbnail takes the user to the actual archived page, giving the user the opportunity to explore more. As an additional feature, Play and Pause buttons along with a loop option is given. Clicking on play and checking the loop option provides the same feel as an animated GIF produced over all these unique thumbnails. Fig. 18 shows a static example of the image slider for www.odu.edu.


Swipe over the images to move through time

Click on an image to view the memento

Datetime: 2010-01-15, 16:02:23

Fig. 18: Slider view of http://www.odu.edu generated by tmvis.

The **Timeline View** arranges the thumbnails according to the datetime. The Timeline view is equipped with zoom, next, previous, next unique, previous unique buttons to easily navigate between the unique and regular thumbnails. All the unique mementos are represented with yellow stripes and the regular ones are represented by gray stripes on the timeline. The Timeline view is based on Timeline Setter library[8], developed by ProPublica. Fig. 19 shows an example of the timeline view for www.odu.edu.

[8] http://propublica.github.io/timeline-setter/

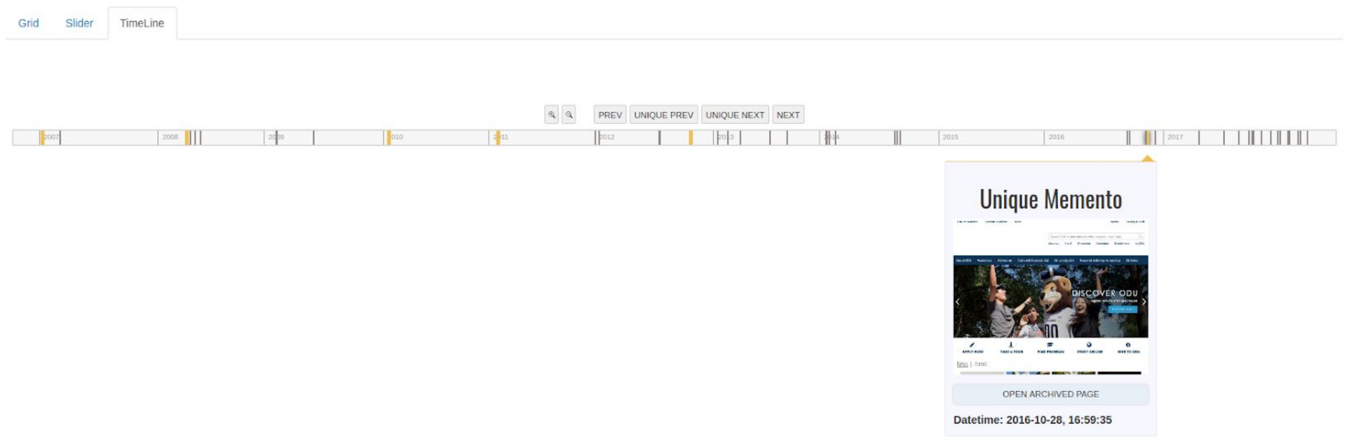


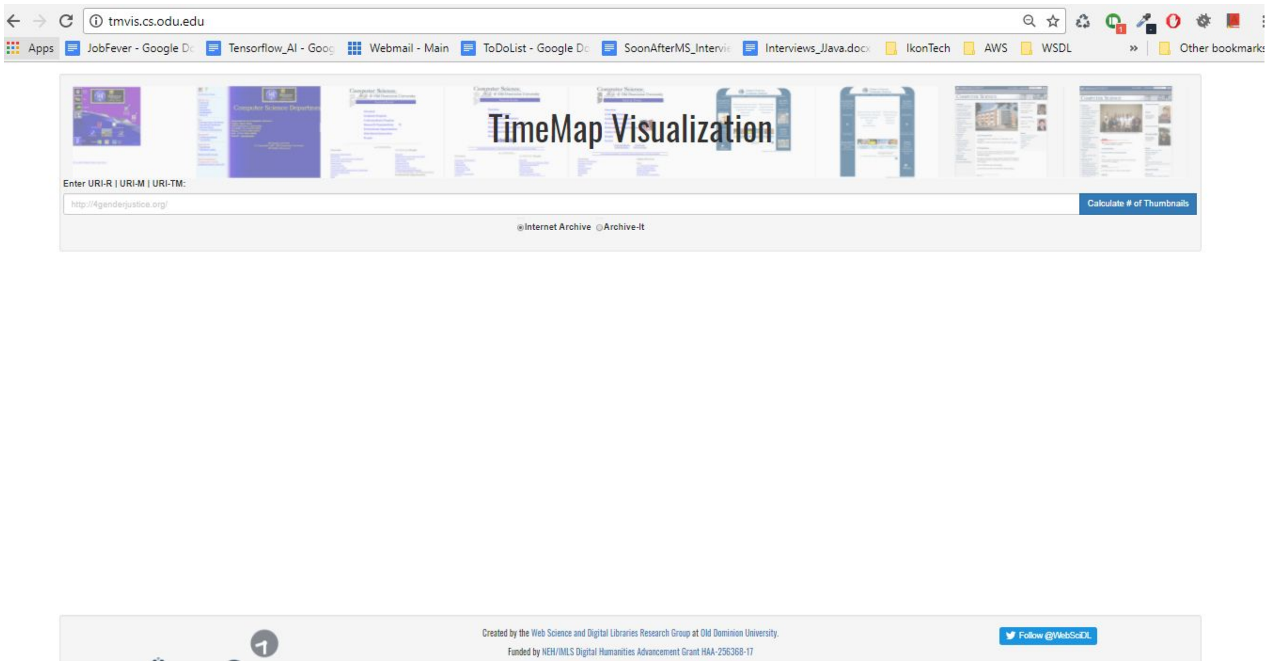
Fig. 19: Timeline view showing summarized http://www.odu.edu generated by tmvis

3.2 System Walkthrough

The task of summarizing the URI happens in two phases from a user's perspective. The first phase shown in Fig. 20, is where the user can request the number of unique thumbnails to summarize the TimeMap for a particular URI. The user can input the URI-R (ex: https://4genderjustice.org/), the URI-M (ex: https://web.archive.org/web/20180326162903/https://4genderjustice.org/), or the URI-T (ex: https://web.archive.org/web/*/https://4genderjustice.org/).

The other input that the user provides to the system in this phase is the source for the archival content, either the Internet Archive or Archive-It. Opting for Archive-It requires the user to pass in collection number as an additional parameter. If nothing is input in here, the value 'all' is considered as the collection number.

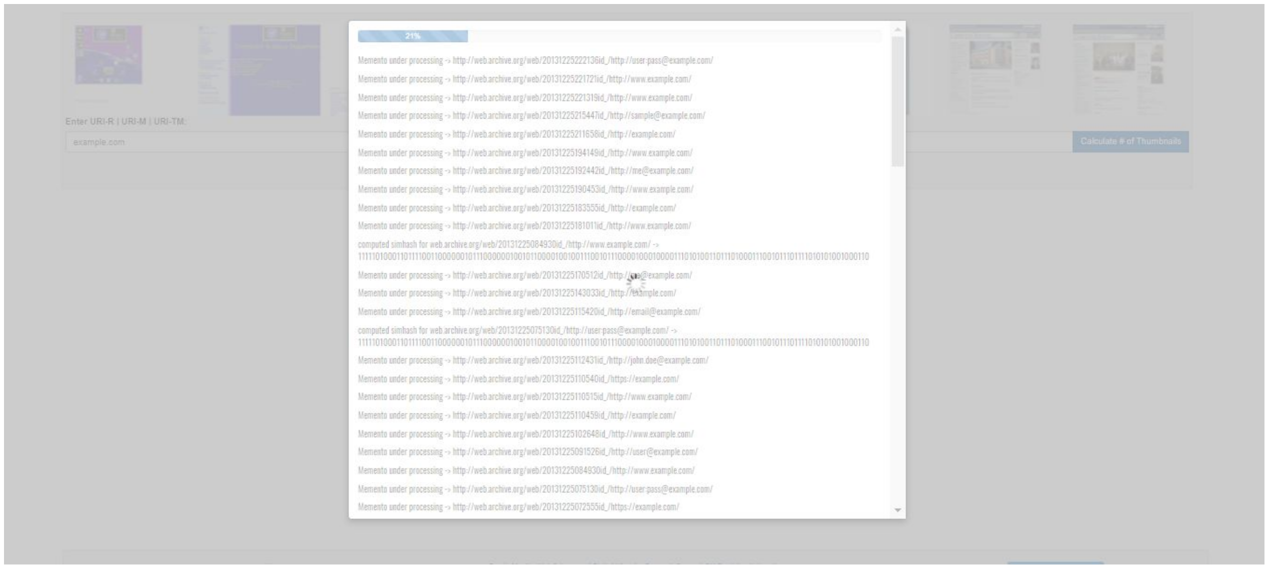
Fig. 20 shows the home page of the application.


tmvis.cs.odu.edu
TimeMap Visualization
Enter URI-R | URI-M | URI-TM:
http://4genderjustice.org/
Calculate # of Thumbnails
Internet Archive
Archive-It
Follow @WebSciDL

Created by the Web Science and Digital Libraries Research Group at Old Dominion University.
Funded by NEH/IMLS Digital Humanities Advancement Grant HAA-256368-17

Fig. 20: Home page of the service tmvis.cs.odu.edu

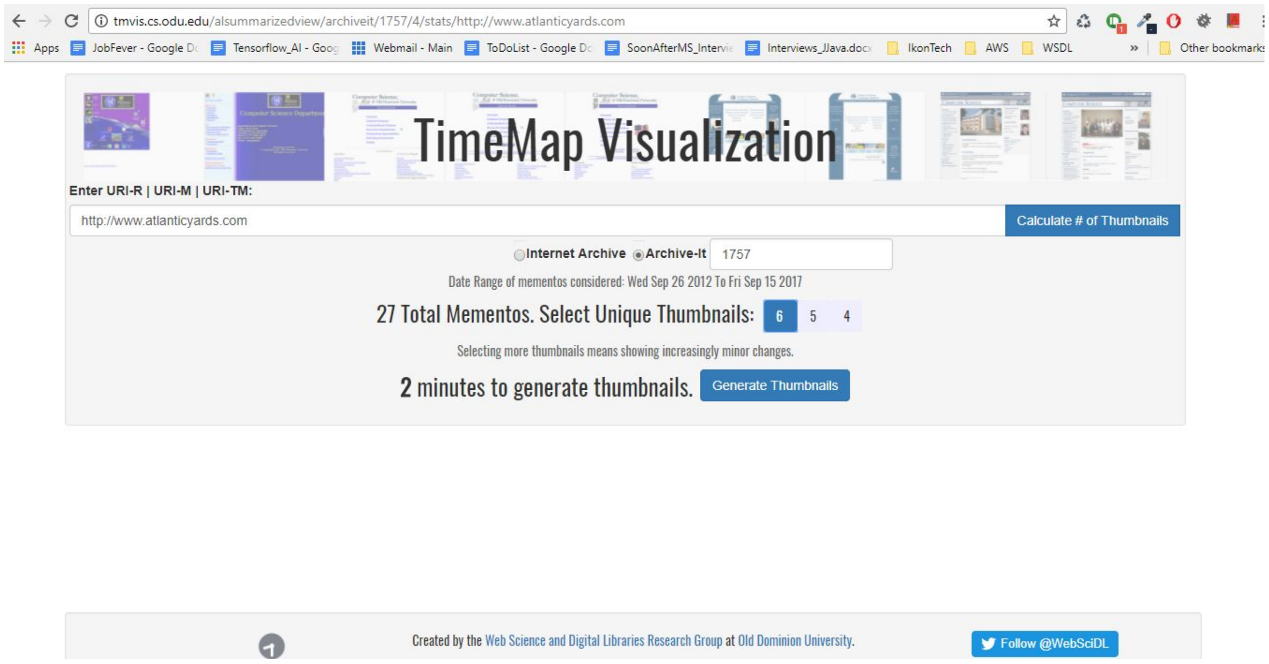
By clicking on the “Calculate # of Thumbnails” button, a request is sent to the server and processing on the server side starts. The user is then notified with the continuous stream of events taking place at the server through a progress window, as in Fig. 21.



```
Memento under processing -> http://web.archive.org/web/20131225222136id_/http://user:pass@example.com/
Memento under processing -> http://web.archive.org/web/20131225221721id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225221319id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225215447id_/http://sample@example.com/
Memento under processing -> http://web.archive.org/web/20131225211658id_/http://example.com/
Memento under processing -> http://web.archive.org/web/20131225194149id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225192442id_/http://me@example.com/
Memento under processing -> http://web.archive.org/web/20131225190453id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225183555id_/http://example.com/
Memento under processing -> http://web.archive.org/web/20131225181011id_/http://www.example.com/
computed simhash for web.archive.org/web/20131225084930id_/http://www.example.com/ ->
1111101000110111100110000001011100000001001011000010010011100101110000100010000111010100110111010001110010111011110101010010001110
Memento under processing -> http://web.archive.org/web/20131225170512id_/http://me@example.com/
Memento under processing -> http://web.archive.org/web/20131225143033id_/http://example.com/
Memento under processing -> http://web.archive.org/web/20131225115420id_/http://email@example.com/
computed simhash for web.archive.org/web/20131225075130id_/http://user:pass@example.com/ ->
1111101000110111100110000001011100000001001011000010010011100101110000100010000111010100110111010001110010111011110101010010001110
Memento under processing -> http://web.archive.org/web/20131225112431id_/http://john.doe@example.com/
Memento under processing -> http://web.archive.org/web/20131225110540id_/https://example.com/
Memento under processing -> http://web.archive.org/web/20131225110515id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225110459id_/http://example.com/
Memento under processing -> http://web.archive.org/web/20131225102648id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225091526id_/http://user@example.com/
Memento under processing -> http://web.archive.org/web/20131225084930id_/http://www.example.com/
Memento under processing -> http://web.archive.org/web/20131225075130id_/http://user:pass@example.com/
Memento under processing -> http://web.archive.org/web/20131225072555id_/https://example.com/
```

Fig. 21: Home page of tmvis.cs.odu.edu with progress window

Once the server side computation of unique number of representative thumbnails is done, the user is presented with the date range of mementos under consideration along with options for choosing the number of unique thumbnails. Fig. 22 shows the output that tmvis has provided after computing the number of unique thumbnails on http://www.atlanticyards.com with Archive-It as the source. Fig. 22 shows that tmvis can summarize the TimeMap of http://www.atlanticyards.com using 6, 5, or 4 thumbnails. Choosing 6 thumbnails will summarize the TimeMap with more similar webpages and hence takes more time to generate. Choosing 4 thumbnails summarizes the TimeMap with more distinct webpages.


tmvis.cs.odu.edu/alsummarizedview/archiveit/1757/4/stats/http://www.atlanticyards.com

TimeMap Visualization

Enter URI-R | URI-M | URI-TM:

http://www.atlanticyards.com

Calculate # of Thumbnails

Internet Archive [✓ Archive-It] 1757

Date Range of mementos considered: Wed Sep 26 2012 To Fri Sep 15 2017

27 Total Mementos. Select Unique Thumbnails: [✓ 6] 5 4

Selecting more thumbnails means showing increasingly minor changes.

2 minutes to generate thumbnails. Generate Thumbnails

Created by the Web Science and Digital Libraries Research Group at Old Dominion University.

Follow @WebSciDL

Fig. 22: tmvis.cs.odu.edu showing the number of unique thumbnails calculated for atlanticyards.com

The second phase of Thumbnail Generation starts when the user chooses the desired number of unique thumbnails and then clicks on the “Generate Thumbnails” button. The server starts processing with Puppeteer in action, rendering each memento and capturing a screenshot. Once the screenshots i.e, thumbnails, are captured upon all the mementos, they are presented to user by building the three visualization widgets as show in the Section 3.1. Fig. 23 shows how the user is presented with the unique thumbnails, with the default tab showing the Grid View. The users can switch between the tabs to view the other two visualization widgets, Image Slider and Timeline view.

Video screencasts of the system demo are available at

1. Calculating # of Thumbnails: https://www.youtube.com/watch?v=N_pjaczn3gk
2. Thumbnail Generation: https://www.youtube.com/watch?v=wdtSsxjW38M

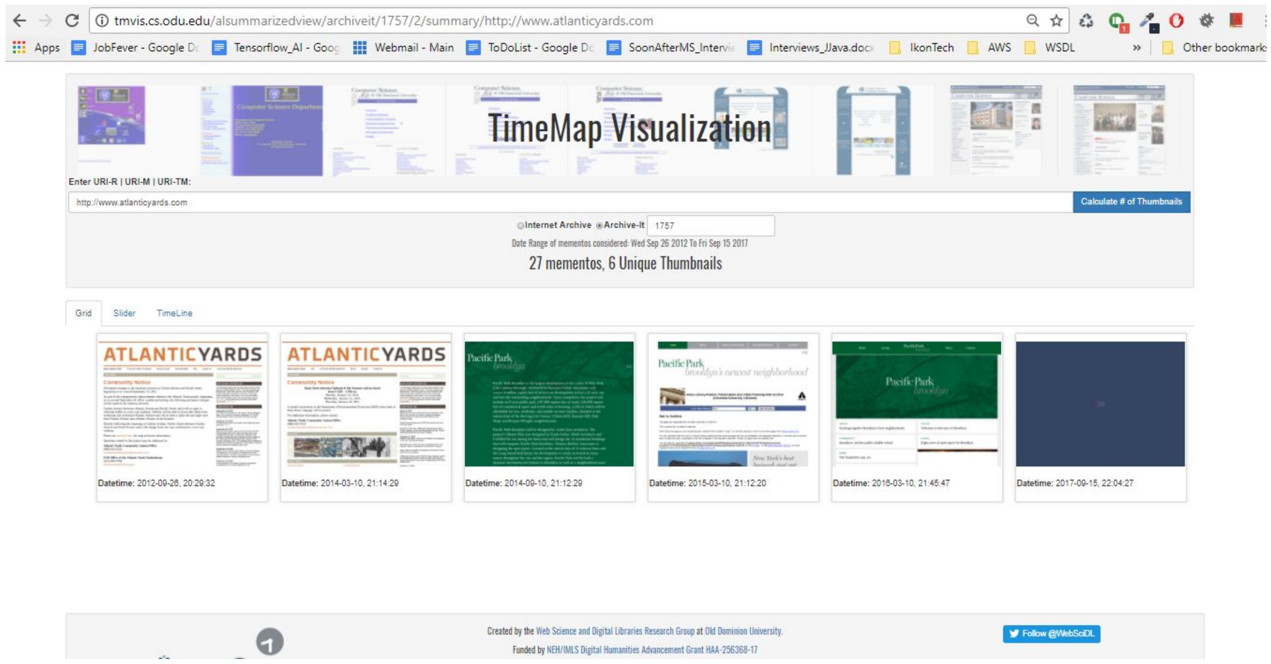

tmvis.cs.odu.edu/alsummarizedview/archiveit/1757/2/summary/http://www.atlanticyards.com
TimeMap Visualization
Enter URI-R | URI-M | URI-TM:
http://www.atlanticyards.com
Calculate # of Thumbnails
Internet Archive
Archive-It
1757
Date Range of mementos considered: Wed Sep 26 2012 To Fri Sep 15 2017
27 mementos, 6 Unique Thumbnails
Grid
Slider
TimeLine
Datetime: 2012-09-26, 20:29:32
Datetime: 2014-03-10, 21:14:29
Datetime: 2014-09-10, 21:12:29
Datetime: 2015-03-10, 21:12:20
Datetime: 2016-03-10, 21:45:47
Datetime: 2017-09-15, 22:04:27
Created by the Web Science and Digital Libraries Research Group at Old Dominion University.
Funded by NEH/IMLS Digital Humanities Advancement Grant HAA-256368-17
Follow @WebSciDL

Fig. 23: tmvis.cs.odu.edu showing the generated thumbnails arranged in grid for atlanticyards.com

4. Technology Stack & Deployment Details

4.1 Technology Stack

The project is developed on Node.js as the backend. The frontend is implemented using HTML, Bootstrap, and JavaScript. In addition, the Timeline Setter library is used to implement the Timeline View. This service is deployed on the WSDL Docker machine maintained by ODU CS.

The service is hosted at http://www.tmvis.cs.odu.edu. The Github repository https://github.com/oduwsdl/tmvis has the entire code base. The README details out all the essentials needed and explains the steps for local deployment, and is included as Appendix A.

4.2 Service Usage

Below are the different ways using which one can use the service.

4.2.1 Using the UI of the webservice

One can summarize the TimeMap of an URI by using the service hosted at http://www.tmvis.cs.odu.edu. Video screencasts of the system demo are as mentioned in Section 3.2.

4.2.2 By constructing an URL

We followed the RESTful design for the tmvis URLs, which makes the construction and sharing easy. Consider that a user wants to summarize the TimeMap of an URI http://4genderjustice.org/ from the collection 1068 of Archive-It using the tmvis hosted

service. The following is the constructed URL of the first phase for knowing the representative mementos count http://tmvis.cs.odu.edu/alsummarizedview/archiveit/1068/4/stats/http://4genderjustice.org, where '**archiveit/1068**' in the URL path represents collection number 1068 from Archive-It as the source. The num
ber '**4**' represents the Hamming distance threshold that decides number of representative mementos to return to the user, and '**stats**' is to notify the service that user is interested in knowing the possible options for representative unique mementos count. The last part of the URL path '**http://4genderjustice.org/**' represents the original URI-R to be summarized by tmvis.

The constructed URL of the second phase is http://tmvis.cs.odu.edu/alsummarizedview/archiveit/1068/4/summary/http://4genderjustice .org. One can notice that **'stats**' is replaced with '**summary**' here and is used to notify the service that user is interested to view the visualizations with the thumbnails created. The user can quickly share this URL with someone else who wants to see the summarized TimeMap. The new user can now view the TimeMap summary of http://4genderjustice.org/ by just pasting the shared tmvis URL in the address bar of the browser.

4.2.3 Run a local instance

Anybody who is interested in running the service locally can do so by following the steps mentioned in project README. Users who wants to summarize popular webpages, which may have quite large TimeMaps, should run the service locally, as the hosted service only considers the latest 5000 mementos for summarization. With the local installation, the user could comment out few lines of code to make the service consider all the mementos of the TimeMap. These changes are outlined in Appendix B.

Also running the service locally gives the user control over the delay that can be provided to the Puppeteer before capturing the screenshot, so that fully rendered thumbnails can be generated. Users who want to extend the functionality of this service by modifying the code base can use the option of running the service on a local Docker[9] container while in development.

4.2.4 CLI tools

In addition to hosted service, a command line tool is also provided through which a user can compute the SimHashes on all the mementos of TimeMap using the option **'os'** (Only SimHash). All the available options are given below. If the option '**os**' is not provided by default all the steps in the process SimHash generation, Hamming distance based filtering, and thumbnails creation (uses PhantomJS) will take place, and a JSON object is returned to the user. Some improvements have to be made here to make these CLI tools more usable. The code base for these CLI tools is at branch **cli** of https://github.com/oduwsdl/tmvis.

The following command shows the way of executing the command line tool and the possible options:

```
> node AlSummarization_OPT_CLI_JSON.js URI-R [--debug] [--hdt 4] [--ia || --ait] [--oes]
[--ci 1068] [-os]

Here is the single line description of what each option stands for
URI-R -> ex: http://4genderjustice.org/
debug -> Run in debug mode
hdt -> Hamming distance Threshold
ia -> The Internet Archive as the source
ait -> Archive-It as the source
oes -> Override Existing Simhashes
ci -> Collection Identifi
os -> Only Simhash
```

[9] https://www.docker.com/what-docker

5. Future Enhancements

Below are plans for future enhancements to tmvis.

5.1 Downloadable Animated GIF

We want to make a downloadable animated GIF as our fourth visualization widget. Currently the animated GIF style of animation on the Image Slider view is implemented via JavaScript. The thumbnails created on the server can be used for making the animated GIF by the usage of any GIF creation libraries. This animated GIF can be made downloadable, which makes the sharing of the summarized TimeMap very easy.

5.2 Embeddable plug and play service

The other idea that we wanted to implement is to make the entire service as an embeddable plug and play resource. The maintainer of a webpage can embed a few lines of code in the webpage's source to get this plug and play resource up and running. Fig. 24 shows a mockup of the image slider widget on the NEH ODH webpage, https://www.neh.gov/divisions/odh.

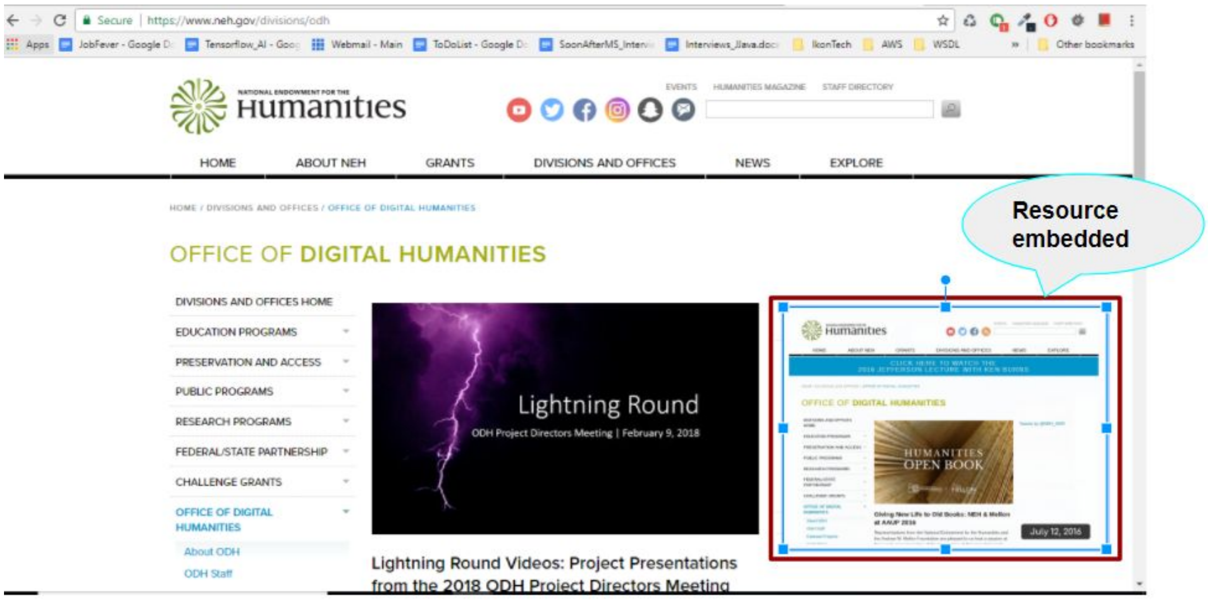


Fig. 24: A picture depicting the image slider widget embedded on the live webpage of https://www.neh.gov/divisions/odh

5.3 User control over representative thumbnails selection

We want to give the user control over selecting the representative mementos, The first control is that user can ask the tmvis service to pick a unique memento one each from each of the years in the TimeMap. The second idea is that we want to provide the user with the feature of selecting and shortlisting the unique mementos that user wants to see in future from the representative mementos returned by tmvis.

6. Conclusion

The goal of this project is to provide a webservice that summarizes the TimeMap of an URI by selecting the unique representative mementos. This service uses the source code from Archive Thumbnails [2] as the base for the server side and the source code from Visualization widgets [3] for the client side. Both the code bases are modified to blend in with each other and merged into a single repository https://github.com/oduwsdl/tmvis. The service is hosted at http://www.tmvis.cs.odu.edu. The server and client in this deployed webservice are not tightly coupled, meaning that the server returns a JSON object as the response, and the client then reads upon this JSON and makes the visualization widgets on the fly. A continuous notification system is also implemented, so the user is better notified with the progress while server computes the results.

References

[1] A. AlSum and M. L. Nelson. 2014. Thumbnail Summarization Techniques for Web Archives. In Proceedings of ECIR, pp. 299-310.

[2] Mat Kelly. 2013. Implementation of Thumbnail Summarization Techniques. https://github.com/machawk1/ArchiveThumbnails

[3] Surbhi Shankar. 2017. Visualizing Thumbnails Of Archived Web Pages,ODU CS Master's Project. http://www.cs.odu.edu/~mweigle/papers/shankar-ms-proj-17.pdf

[4] Herbert Van de Sompel, Michael L. Nelson, and Robert Sanderson. 2013. HTTP framework for time-based access to resource states – Memento, Internet RFC 7089. http://tools.ietf.org/html/rfc7089

[5] Herbert Van de Sompel, Michael L. Nelson, Robert Sanderson, Lyudmila L. Balakireva, Sco Ainsworth, and Harihar Shankar. 2009. Memento: Time Travel for the Web. Technical Report arXiv:0911.1112

[6] Michele C. Weigle. 2017. Visualizing Webpage Changes Over Time - new NEH Digital Humanities Advancement Grant.
http://ws-dl.blogspot.com/2017/10/2017-10-16-visualizing-webpage-changes.html

Appendix A

The following is the README the repository https://github.com/oduwsdl/tmvis, as of May 1, 2018.

Timemap Visualization

A web service for TimeMap visualization based on Mat's (@machawk1) https://github.com/machawk1/ArchiveThumbnails which itself is an implementation of Ahmed AlSum's 2014 ECIR paper titled "Thumbnail Summarization Techniques for Web Archives" for the Web Archiving Incentive Program for Columbia University Libraries' grant, "Visualizing Digital Collections of Web Archives".

Requirements

Node.js is required to run the service. Once Node is installed, the packages required to use the service can be installed by running npm install -g in the root of the project directory.

Running

To execute the code, run `node tmvis.js`.

To query the server instance generated using your browser visit `http://localhost:3000/alsummarizedtimemap/archiveit/1068/4/[stats | summary]/http://4genderjustice.org/`, which has the attributes path as `primesource/ci/hdt/role/URI-R` substitute the URI-R to request a different webpage summarization. The additional parameters of role is used to specify the values '`stats`' or '`summary`',

stats: for getting the no of unique mementos

summary: to get the the unique mementos along with the screenshots captured

`ci` is used to specify the collection identifier if not specified the argument **'all'** is used, `primesource` gets the value of 'archiveIt' or 'internetarchive' as to let the service know which is the primary source.

Example URIs

- `http://localhost:3000/alsummarizedtimemap/archiveit/1068/4/stats/http://4genderjustice.org/`
- `http://localhost:3000/alsummarizedtimemap/archiveit/1068/4/summary/http://4genderjustice.org/`

Running as a Docker Container (Non development mode: Recommended for naive users)

Follow the following steps:

```
$ git clone https://github.com/oduwsdl/tmvis.git
$ cd tmvis
$ docker image build -t timemapvis .
$ docker container run --shm-size=1G -it --rm -p 3000:3000 timemapvis node
tmvis.js
```

Running as a Docker Container (experimental)

Running the server in a Docker container can make the process of dependency management easier. The code is shipped with a Dockerfile to build a Docker image that will run the service when started. This document assumes that you have Docker setup already, if not then follow the official guide.

Building Docker Image

Clone the repository and change working directory (if not already) then build the image.

```
$ git clone https://github.com/oduwsdl/tmvis.git
$ cd tmvis
$ docker image build -t timemapvis .
```

In the above command `timemapvis` is the name of the image which can be anything, but the same needs to be used when running the container instance.

Running Docker Container

```
docker run -it --rm timemapvis bash
```

In another terminal

```
cd tmvis
docker cp (CONTAINER ID CREATED ABOVE):/app/node_modules/ ./

docker run --shm-size=1G -it --rm -v "$PWD":/app -p 3000:3000 --user=$(id
-u):$(id -g) timemapvis bash
node tmvis.js
```

In the above command the container is running in detached mode and can be accessed from outside on port `3000` at http://localhost:3000/. If you want to run the service on a different port, say `80` then change `-p 3000:3000` to `-p 80:3000`.

In order to persist generated thumbnails, mount a host directory as a volume inside the container by adding `-v /SOME/HOST/DIRECTORY:/app/assets/screenshots` flag when running the container.

Container is completely transparent from the outside and it will be accessed as if the service is running in the host machine itself.

In case if you want to make changes in the `tmvis` code itself, you might want to run it in the development mode by mounting the code from the host machine inside the container so that changes are reflected immediately, without requiring an image rebuild. Here is a possible workflow:

```
$ git clone https://github.com/oduwsdl/tmvis.git
$ cd tmvis
$ docker image build -t timemapvis .
$ docker container run --shm-size=1G -it --rm -v "$PWD":/app --user=$(id
-u):$(id -g) timemapvis npm install
$ docker container run --shm-size=1G -it --rm -v "$PWD":/app -p 3000:3000
--user=$(id -u):$(id -g) timemapvis
```

Once the image is built and dependencies are installed locally under the `node_modules` directory of the local clone, only the last command would be needed for continuous development. Since the default container runs under the `root` user, there might be permission related issues on the `npm install` step. If so, then try to manually create the `node_modules` directory and change its permissions to world writable (`chmod -R a+w node_modules`) then run the command to install dependencies again.

Regarding License

Though GPL Licensing was used for base (https://github.com/machawk1/ArchiveThumbnails) of this repository, we have chosen the MIT license[10]. This is changed with the permission from the original author, @machawk1.

[10] https://en.wikipedia.org/wiki/MIT_License

Usage of the service

Running this service gives provides an user with the array of JSON object as the response (webservice model), which then has to be visualized with the UI tool deployed at http://tmvis.cs.odu.edu/ for which the code is available at https://github.com/oduwsdl/tmvis/ under public folder

Request format (Role -> stats)

```
curl -il
http://localhost:3000/alsummarizedtimemap/archiveIt/1068/4/stats/http://4gender
justice.org/

Mapping of attributes of URI to the values are as follows:
  primesource -> archiveIt
  hammingdistance -> 4
  role -> stats
  collection Identifier -> 1068
  URI-R under request -> http://4genderjustice.org/
```

Response format

```
{
  "totalmementos": 21,
  "unique": 2,
  "timetowait": 0
}
```

Request format (Role -> summary)

```
curl -il
http://localhost:3000/alsummarizedtimemap/archiveIt/1068/4/summary/http://4gende
rjustice.org/
```

```
Mapping of attributes of URI to the values are as follows:
  primesource -> archiveIt
  hammingdistance -> 4
  role -> summary
  collection Identifier -> 1068
  URI-R under request -> http://4genderjustice.org/
```

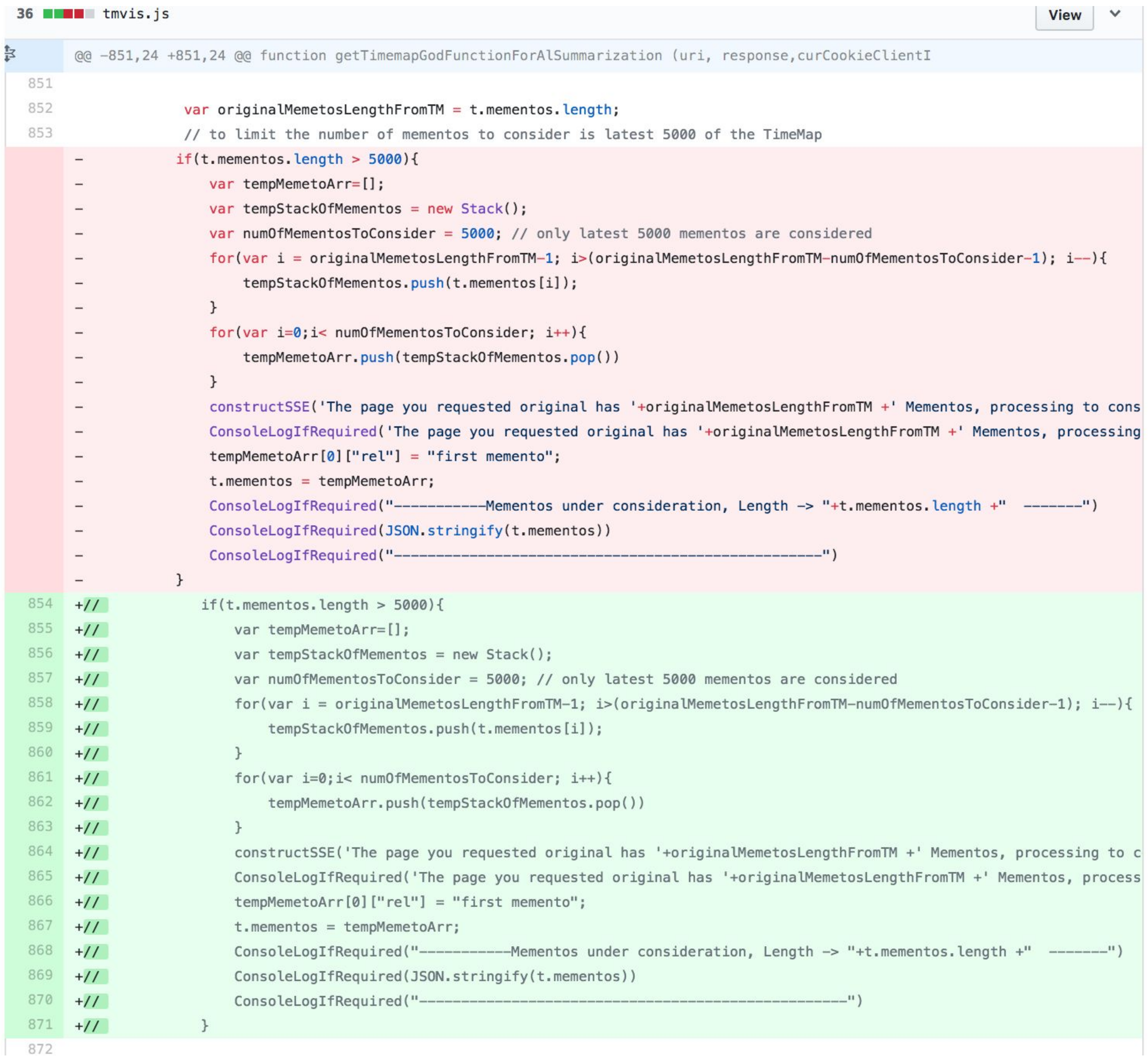
Response format

```
[
  {
    "timestamp": 1435787801,
    "event_series": "Thumbnails",
    "event_html":
'http://localhost:3000/static/timemapSum_httpwaybackarchiveitorg106820150701215641http4genderjusticeorg.png',
    "event_date": "Aug. 01, 2015",
    "event_display_date": "2015-07-01, 21:56:41",
    "event_description": "",
    "event_link":
"http://wayback.archive-it.org/1068/20150701215641/http://4genderjustice.org/"
  },
  {
    "timestamp": 1435789960,
    "event_series": "Non-Thumbnail Mementos",
    "event_html": 'http://localhost:3000/static/notcaptured.png',
    "event_html_similarto":
'http://localhost:3000/static/timemapSum_httpwaybackarchiveitorg106820150701215641http4genderjusticeorg.png',
    "event_date": "Aug. 01, 2015",
    "event_display_date": "2015-07-01, 22:32:40",
    "event_description": "",
    "event_link":
"http://wayback.archive-it.org/1068/20150701223240/http://4genderjustice.org/"
```

```
  },....
]
```

Appendix B

The following screenshot of the code snippet shows the modification needed in the code to consider all the mementos of TimeMap which was limited to last 5000 mementos before. The same can be accessed using the link https://github.com/oduwsdl/tmvis/commit/3a47650700cd459cf28182b5572c411f4b482bac



```
36 tmvis.js                                                                                    View
@@ -851,24 +851,24 @@ function getTimemapGodFunctionForAlSummarization (uri, response,curCookieClientI

             var originalMemetosLengthFromTM = t.mementos.length;
             // to limit the number of mementos to consider is latest 5000 of the TimeMap
-           if(t.mementos.length > 5000){
-               var tempMemetoArr=[];
-               var tempStackOfMementos = new Stack();
-               var numOfMementosToConsider = 5000; // only latest 5000 mementos are considered
-               for(var i = originalMemetosLengthFromTM-1; i>(originalMemetosLengthFromTM-numOfMementosToConsider-1); i--){
-                   tempStackOfMementos.push(t.mementos[i]);
-               }
-               for(var i=0;i< numOfMementosToConsider; i++){
-                   tempMemetoArr.push(tempStackOfMementos.pop())
-               }
-               constructSSE('The page you requested original has '+originalMemetosLengthFromTM +' Mementos, processing to cons
-               ConsoleLogIfRequired('The page you requested original has '+originalMemetosLengthFromTM +' Mementos, processing
-               tempMemetoArr[0]["rel"] = "first memento";
-               t.mementos = tempMemetoArr;
-               ConsoleLogIfRequired("-----------Mementos under consideration, Length -> "+t.mementos.length +"  -------")
-               ConsoleLogIfRequired(JSON.stringify(t.mementos))
-               ConsoleLogIfRequired("----------------------------------------------------")
-           }
+//           if(t.mementos.length > 5000){
+//               var tempMemetoArr=[];
+//               var tempStackOfMementos = new Stack();
+//               var numOfMementosToConsider = 5000; // only latest 5000 mementos are considered
+//               for(var i = originalMemetosLengthFromTM-1; i>(originalMemetosLengthFromTM-numOfMementosToConsider-1); i--){
+//                   tempStackOfMementos.push(t.mementos[i]);
+//               }
+//               for(var i=0;i< numOfMementosToConsider; i++){
+//                   tempMemetoArr.push(tempStackOfMementos.pop())
+//               }
+//               constructSSE('The page you requested original has '+originalMemetosLengthFromTM +' Mementos, processing to c
+//               ConsoleLogIfRequired('The page you requested original has '+originalMemetosLengthFromTM +' Mementos, process
+//               tempMemetoArr[0]["rel"] = "first memento";
+//               t.mementos = tempMemetoArr;
+//               ConsoleLogIfRequired("-----------Mementos under consideration, Length -> "+t.mementos.length +"  -------")
+//               ConsoleLogIfRequired(JSON.stringify(t.mementos))
+//               ConsoleLogIfRequired("----------------------------------------------------")
+//           }
```

Fig. 25: A picture depicting the modifications to be done to consider all the mementos into account.