SUPPORTING ACCOUNT-BASED QUERIES FOR ARCHIVED INSTAGRAM POSTS

by

Himarsha R. Jayanetti
B.E. May 2017, Gujarat Technological University, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
May 2023

Approved by:

Michele C. Weigle (Director)

Michael L. Nelson (Member)

Faryaneh Poursardar (Member)

ABSTRACT

SUPPORTING ACCOUNT-BASED QUERIES FOR ARCHIVED INSTAGRAM POSTS

Himarsha R. Jayanetti
Old Dominion University, 2023
Director: Dr. Michele C. Weigle

Social media has become one of the primary modes of communication in recent times, with popular platforms such as Facebook, Twitter, and Instagram leading the way. Despite its popularity, Instagram has not received as much attention in academic research compared to Facebook and Twitter, and its significant role in contemporary society is often overlooked. Web archives are making efforts to preserve social media content despite the challenges posed by the dynamic nature of these sites. The goal of our research is to facilitate the easy discovery of archived copies, or mementos, of all posts belonging to a specific Instagram account in web archives. We proposed two approaches to support account-based queries for archived Instagram posts. The first approach uses existing technologies in the Internet Archive by using WARC revisit records to incorporate Instagram usernames into the `WARC-Target-URI` field in the WARC file header. The second approach involves building an external index that maps Instagram user accounts to their posts. The user can query this index to retrieve all post URLs for a particular user, which they can then use to query web archives for each individual post. The implementation of both approaches was demonstrated, and their advantages and disadvantages were discussed. This research will enable web archivists to make informed decisions on which approach to adopt based on practicality and unique requirements for their archives.

I dedicate this thesis to my beloved Amma, Badra Jayanetti, whose constant inspiration and unwavering support have been the guiding light throughout my life.

ACKNOWLEDGMENTS

I want to start by expressing my gratitude to Drs. Michele C. Weigle and Michael L. Nelson, who served as my advisors during my master's program. My thesis proposal would never have materialized without their guidance and support. I would like to express my gratitude to Dr. Faryaneh Poursardar for providing valuable feedback throughout the process. My heartfelt gratitude goes to the other faculties in the Web Science and Digital Libraries (WS-DL) research group, Drs. Sampath Jayarathna, Jian Wu, Vikas Ashok, and the chair of the Computer Science department Dr. Ravi Mukkamala, for always being willing to share their knowledge and offer advice. My deepest appreciation goes out to the students in the WS-DL group, whose invaluable assistance in collecting the data was instrumental to the success of my research. I would like to acknowledge Mark Graham, the director of the Wayback Machine, and Dr. Sawood Alam, a web and data scientist at the Wayback Machine and an alumnus of WS-DL ODU, for generously sharing the access log data from the Internet Archive's Wayback Machine. Without the help of my family (Amma, Thaththa, Nanda, Akka, Ayya, and my nieces Tiana and Taliyah), who kept me sane during stressful times my academic journey would not have been successful. Their affection and support have encouraged and uplifted me throughout my undergraduate and graduate studies. I am grateful to my extended family (Amma, Appa, Akka, Athan, Maya, and Arya) who have shown immense interest and support in my academic pursuits. Finally, I am deeply indebted to my husband, Skanda Siva, for being a loving and supportive partner, especially throughout the long hours I put in at work. I am truly thankful for your constant support.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Social media is a popular way in which people interact with one another. Popular examples of social media websites are Facebook,[1] Twitter,[2] and Instagram[3] (IG, or Insta), each of which offer specific features. Facebook and Twitter are social networking sites where you can publish status updates, photos, and videos. Instagram is a media-sharing network solely focused on sharing user-generated content like photos and videos. As of January 2023, Facebook has almost 3 billion monthly active users [1], Twitter 450 million [2], and Instagram 2 billion [3].

Compared to the two other social media giants, Facebook and Twitter, Instagram is understudied in academic research compared to its popularity. As of January 22, 2023, Google Scholar[4] returns approximately 7.22 million hits for "facebook", 8.01 million hits for "twitter", and 3.25 million hits for "instagram". This results in an "active users/Google Scholar hits" ratio of 56:1 for Twitter, 405:1 for Facebook, and 615:1 for Instagram. Compared to Facebook and Twitter, Instagram has a higher ratio (1.5 times higher than Facebook and 10 times higher than that Twitter) showing that Instagram is under-represented in academic publications. But in recent years, we have noticed that Instagram has become increasingly popular among academic scholars. For example, in a prior study [4], we reported that as of February 11, 2021, there were 1.54 million hits for "instagram" in Google Scholar, showing that the new data reflects a doubling in the number of hits for Instagram while the others just increased slightly.

Instagram is understudied for many different reasons. First, unlike the Twitter Developer API [5], Instagram's Graph API [6] and Basic Display API [7] are designed to give businesses and content creators access to information about their own accounts rather than to enable researchers to analyze other accounts. Secondly, because Instagram does not have a built-in sharing feature, it is difficult to share content with other Instagram accounts, making it challenging to track such engagements. In contrast, Facebook's "share" feature and Twitter's "retweet" feature allows researchers to analyze user engagement and information-propagation networks [8, 9, 10, 11, 12]. Another possible cause is the fact that text is considerably easier to analyze than image or video content. While Facebook and Twitter both allow users to post tweets with only text as well as upload both images and videos, Instagram only permits posts that include a picture or a video (no

[1] https://www.facebook.com/

[2] https://twitter.com/

[3] https://www.instagram.com/

[4] https://scholar.google.com/

posts with only text). Finally, the under-representation of Instagram in academic studies might be because Instagram's involvement in numerous operations has gone mostly unnoticed. For example, while social media sites like Facebook and Twitter were known platforms for disinformation/misinformation campaigns, Instagram also played a significant role in such efforts, which is often overlooked. For example, in December 2018, the Senate Intelligence Committee[5] released a report by New Knowledge [13] on the strategies used by the Russian Internet Research Agency (IRA),[6] the group that the Special Counsel's Office accused of engaging in a conspiracy to defraud the United States. The report stated that *"In 2017, as media covered their Facebook and Twitter operations, the IRA shifted much of its activity to Instagram"* and *"Instagram was perhaps the most effective platform for the Internet Research Agency"*.

This shows that the digital content that is present today (even in the form of social media) is the historical evidence of tomorrow. Because of this, preserving social media platforms like Facebook, Twitter, and Instagram have gained popularity. Several advances have been made in preserving web content over the years. Social media sites are complex because of their dynamic nature, which makes it challenging to preserve them while safeguarding the integrity and identity of data for interested parties. If this web content is not preserved properly, future historians and scholars will not be able to access crucial information with historical worth.

1.1 PROBLEM

The aim of our research is to allow easy discovery of archived copies, or mementos, for all the posts of a particular Instagram user account in web archives. For instance, Bragg et al. measured the replayability and quality of archived Instagram account pages of a group of conspiracy theorists (known as the "Disinformation Dozen") and several health authorities [14, 15]. If it was possible to discover all the posts for these interested accounts in their study, Bragg et al. could have explored all the archived Instagram post pages and not just account pages.

To illustrate the difficulties one would have while attempting to find archived Instagram posts that belong to a particular user account, we introduce Maya, a researcher studying conspiracy theories and conspiracy theorists by studying known social media accounts. We will use the Instagram account of a well-known conspiracy theorist Naomi R. Wolf to illustrate the difficulties one would have while attempting to find archived Instagram posts that belong to a particular user account. We will also use her Twitter account to compare and contrast the challenges posed on Instagram. Twitter suspended the account of Naomi Wolf (`@naomirwolf`)[7] after tweeting anti-vaccine mis-

[5] `https://www.intelligence.senate.gov/`

[6] `https://en.wikipedia.org/wiki/Internet_Research_Agency`

[7] `https://twitter.com/naomirwolf/`

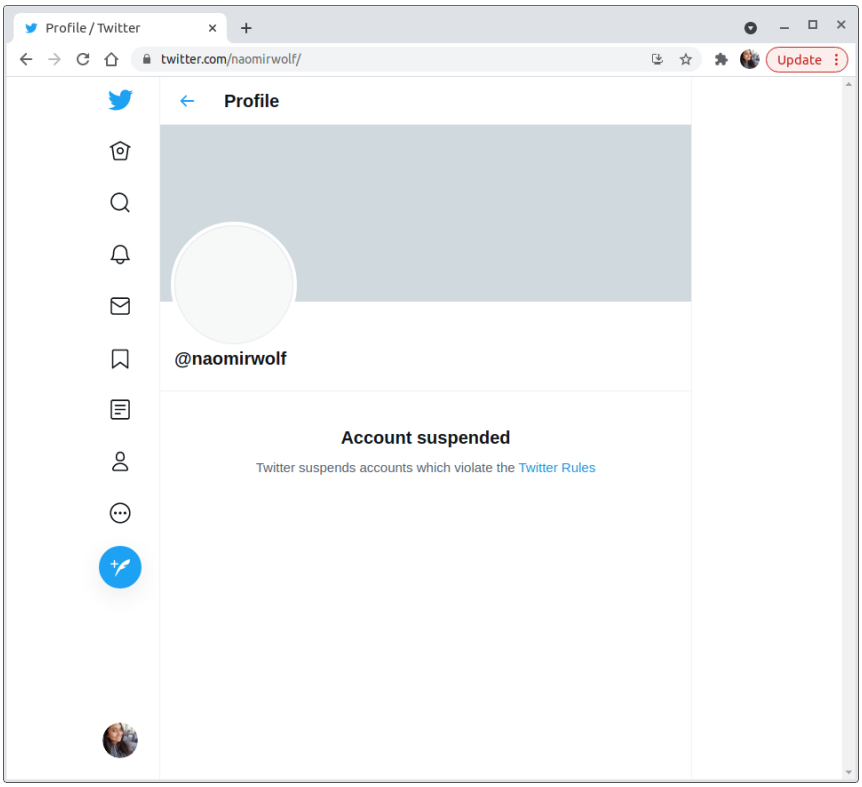

Fig. 1. A screenshot of the suspended Twitter account of Naomi Wolf (`@naomirwolf`) on the live web as captured on October 25, 2021 - https://twitter.com/naomirwolf/.

information [16] in June 2021. Figure 1 shows a screenshot of her suspended Twitter account. Because of the suspension, the `(@naomirwolf)` Twitter account is no longer on the live web and one would have to rely on web archives to access mementos of Naomi Wolf's Twitter account page and tweets.

Maya used the Wayback Machine user interface to look for archived pages in the Internet Archive. As shown in Figure 2, by using the URL `https://twitter.com/naomirwolf/` as a lookup, Maya found out that the account page has been archived only 490 times. She also used `https://twitter.com/naomirwolf/status/*` (with a star at the end indicating match everything that begins with the URL prefix) to look for mementos of individual tweets and found out that more than 100,000 archived copies exist for the tweet pages (Figure 3). This shows how there are not nearly as many mementos available for her account page (top-level page), which is mostly what users know how to find, as for other tweet pages combined. Therefore, we can use this method to discover the links to the tweets that would otherwise be hard to find.

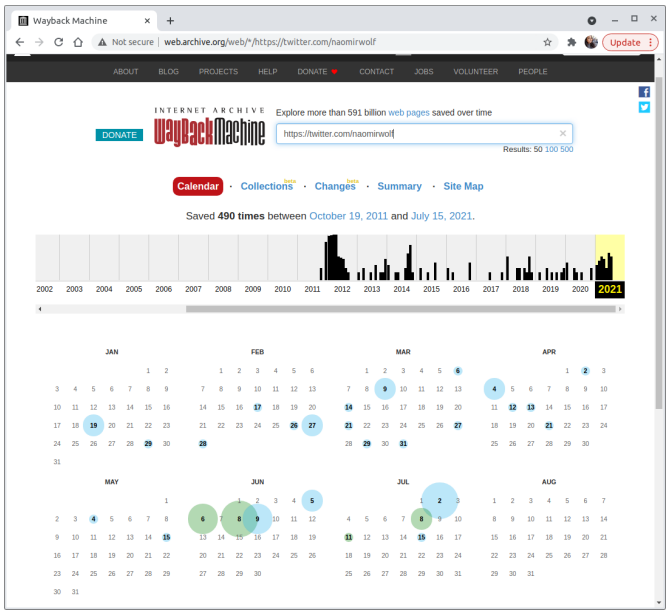

Fig. 2. The Wayback Machine 2021 calendar page for the Twitter account page of Naomi Wolf http://web.archive.org/web/*/https://twitter.com/naomirwolf/.

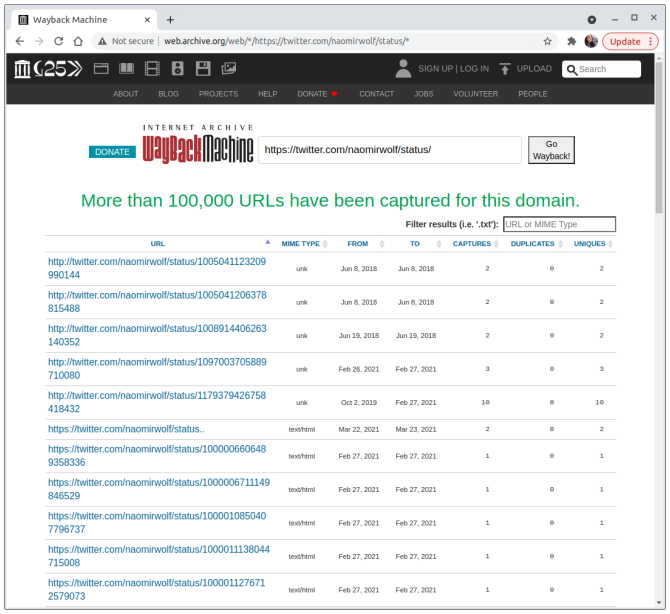

Fig. 3. Mementos for the individual tweets of Naomi Wolf using the Wayback Machine user interface - http://web.archive.org/web/*/https://twitter.com/naomirwolf/status/*.

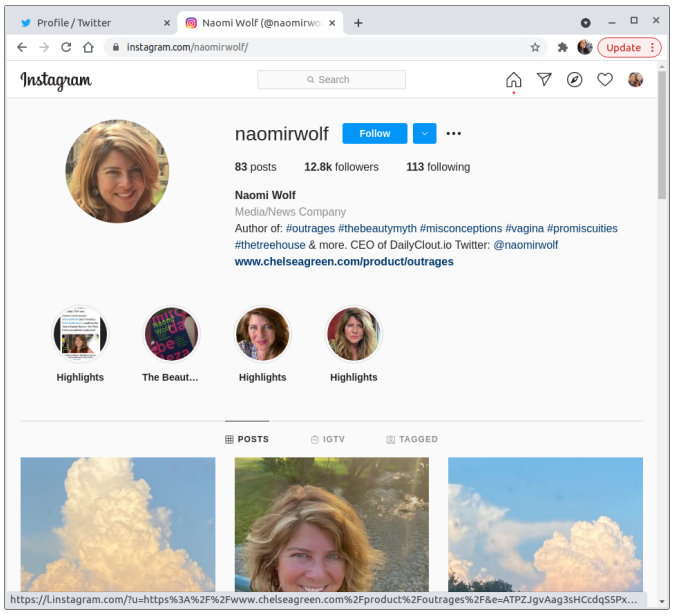

Fig. 4. The Instagram account of Naomi Wolf (`(@naomirwolf)`) - https://www.instagram.com/naomirwolf/

Similarly, Maya decides to do some research on Naomi Wolf's Instagram account (Figure 4). As we can see in Figure 5, there are only six mementos of the account page of Naomi Wolf's Instagram account. Maya thinks she will use the same method she did for finding archived pages for individual tweets to find archived pages for individual posts on Naomi Wolf's Instagram account. However, Maya is not sure which URL prefix to use. That is when she noticed that although Twitter has the account name of the user account associated (`@naomirwolf`) in the tweet ID URL (`https://twitter.com/naomirwolf/status/*`), Instagram does not follow the same URL structure. Figure 6 shows the results Maya gets if she uses the URL prefix `https://www.instagram.com/naomirwolf/*` in the Wayback Machine UI. There are 50 mementos and they only show mementos for different language variations of the Instagram account page of `@naomirwolf`. When Maya looked into one of Naomi Wolf's post URLs (for example, `https://www.instagram.com/p/BW8BCZUDLkP/`), she found that it did not have a username in it. This illustrates how Instagram post URLs are completely opaque. By only looking at a post URL, we cannot say which user owns the post without dereferencing the URL. This means that if a web archive user like Maya wants to find mementos of an Instagram post, they should know the exact URL of the post to query for. For the same reason, Maya cannot easily type in a prefix to the search bar to identify mementos for all of Naomi Wolf's Instagram posts.

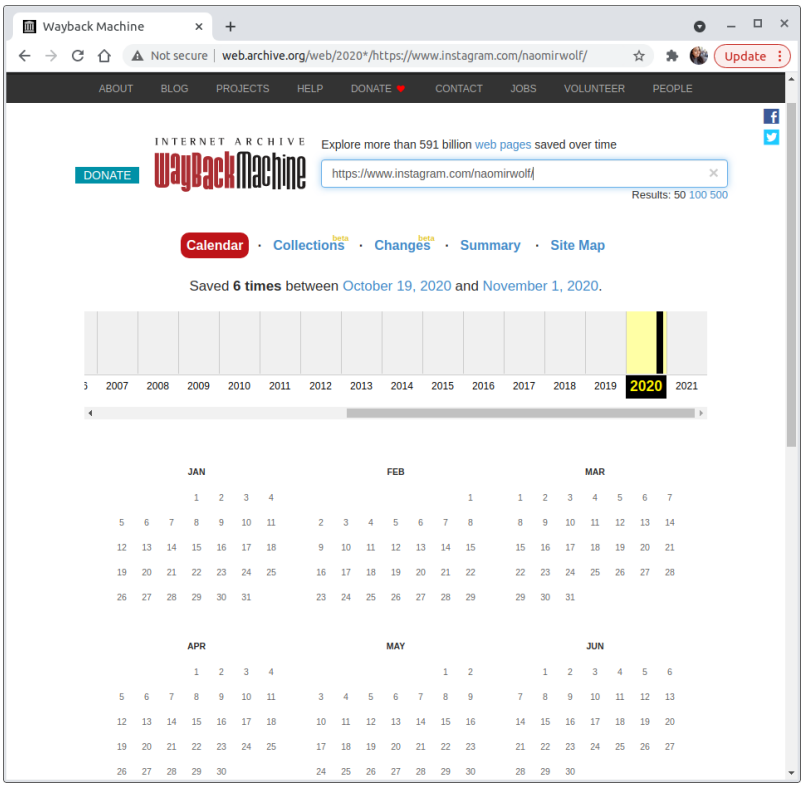

Fig. 5. Wayback Machine 2020 calendar page for the Instagram account page of Naomi Wolf - http://web.archive.org/web/*/https://www.instagram.com/naomirwolf/

In this thesis, we explore approaches to address the following research question: How can we facilitate the discovery of all of a user's archived Instagram posts? These approaches will be discussed in detail in Chapter 4, but we provide an overview here. One such approach is by using the already existing technologies in the Internet Archive to create a mapping between an Instagram user and that user's posts. This way a URL with the username (`https://www.instagram.com/naomirwolf/p/BW8BCZUDLkP/`) and without the username (`https://www.instagram.com/p/BW8BCZUDLkP/`) would both point to the same memento. This would make the prefix search (`https://www.instagram.com/naomirwolf/p/*`) automatically work. The other approach would be by building a separate external index that has a mapping between Instagram user accounts and their posts. When a user wants to get all post URLs for a particular Instagram user, they can first query the index, which provides all posts that belong to a username. From this index, the user can acquire all post URLs with which they can query the Internet Archive for each individual post separately. This index will be slowly built by different methods. Over time, we would be able to get many (not all) post and user account data which would help us build a near-complete index.

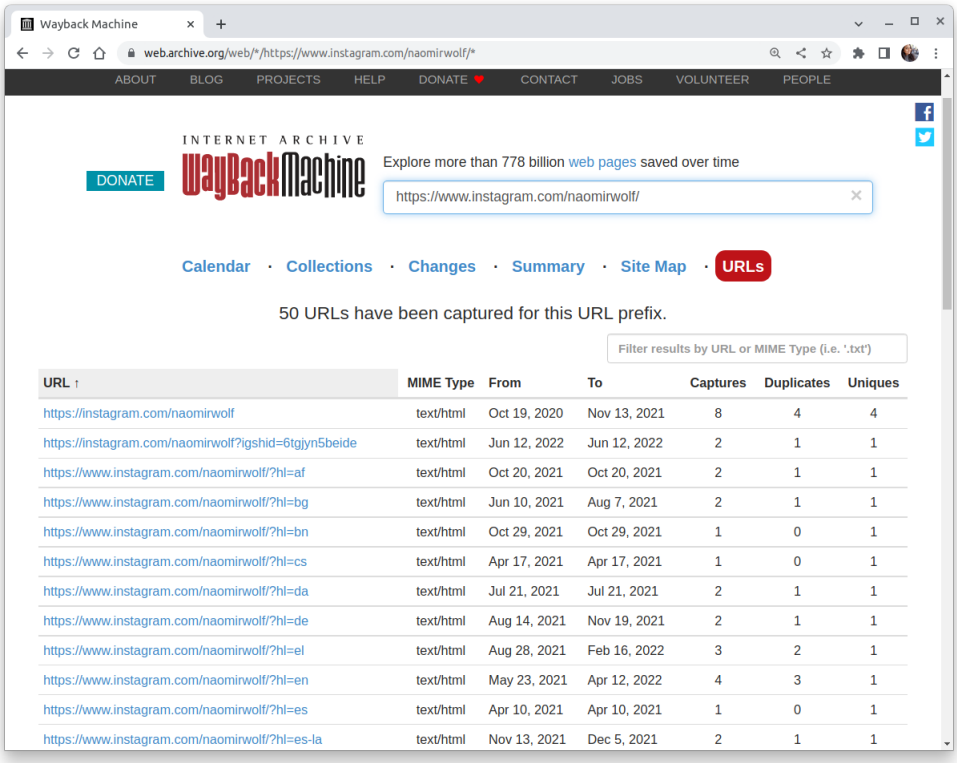

Fig. 6. Mementos for the URL prefix search matching the prefix of the account page URL (https://www.instagram.com/naomirwolf/).
http://web.archive.org/web/*/https://www.instagram.com/naomirwolf/*

1.2 CONTRIBUTIONS

This thesis makes the following research contributions:

1. Highlighting the challenges of accessing Instagram content, specifically posts belonging to a particular user, through web archives.

2. Proposing an approach that involves using WARC revisit records to add Instagram usernames to the `WARC-Target-URI`, enabling users to perform a prefix search and discover Instagram posts belonging to a specific user.

3. Proposing an approach for creating a secondary index that associates user accounts with their post URLs. With this approach, users can initially search the index to locate all posts belonging to a particular user and then query the web archive in the usual manner.

1.3 THESIS ORGANIZATION

This thesis has six chapters and they are organized as below. In this chapter (Chapter 1) we provided an introduction to my research by introducing the problem and by listing approaches and contributions of this thesis. Chapter 2 provides the reader with a foundational understanding of the concepts and technologies required to aid in understanding the rest of the thesis. This chapter covers an overview of the World Wide Web, Uniform Resource Identifiers (URIs), and web archiving basics such as the Memento protocol, WARC files, and CDX server API, and briefly mentions the transient nature of social media accounts. Chapter 3 is dedicated to exploring the existing studies conducted in the field of research related to social media and web archiving, and their relevance to our work. Chapter 4 outlines the various approaches proposed to facilitate the retrieval of archived Instagram posts belonging to a specific user from web archives. It also includes an initial implementation of these approaches and the data used for this purpose. Chapter 5 explores the future of this research by discussing the next steps for some of the approaches outlined in Chapter 4. Furthermore, it highlights the necessity of exploring other aspects of social media archiving and web archiving that have not been addressed in this thesis. Chapter 6 brings together all the discussed concepts, approaches, and implementations to demonstrate the feasibility of supporting queries to find all posts of an Instagram user account available in web archives. Finally, Appendices A through J contain supplementary materials, such as code snippets and examples.

CHAPTER 2

BACKGROUND

This chapter provides a brief background to the concepts and technologies that are required to comprehend the rest of this thesis.

2.1 THE WORLD WIDE WEB AND UNIFORM RESOURCE IDENTIFIERS

Tim Berners-Lee, who developed the World Wide Web in 1989, created the core technologies that support the Web, including Hypertext Markup Language (HTML) [17] to create web pages, Hypertext Transfer Protocol (HTTP) [18] to connect web clients and servers, and Uniform Resource Identifier (URI) [19] to find a resource. Anything that can be identified using a Uniform Resource Identifier (URI), such as web pages, images, and product catalogs, is considered to be a resource [20]. A URI constitutes multiple segments like the scheme, domain, port number, path, query, etc. In the URI `https://www.example.com:8042/search?foo=bar#baz`, `https` is the scheme, `example.com` is the domain, `8042` is the port number, `/search` is the path, and `foo=bar` is the query, and `baz` is the fragment. The most common URI is a Uniform Resource Locator (URL), also known as the web address. It is used to properly identify and locate websites or other resources connected to the web.

While a URI serves as a unique identifier, enabling the differentiation of one resource from another, a resource may have one or more representations. For instance, a document may have both PDF and HTML representations. People tend to infer resource attributes by looking at a URI that identifies it, but the web is made to communicate resource information status using representations rather than identifiers. The World Wide Web Consortium in "Architecture of the World Wide Web, Volume One" stated that by looking at a resource's URI, one should not infer the properties of the referenced resource or establish the type of a resource representation [21]. The process of obtaining a representation from a URI is called dereferencing. The most widely used protocol for obtaining representations of resources from URIs is HTTP, which employs a system of requests and responses. Requests are generated by a User Agent and transmitted to the server, which then sends back the requested representation [18]. These requests are usually initiated by the user through a client tool such as a web browser. These requests use one of several available methods such as GET, HEAD, POST, etc. [22], that web clients can use to make HTTP requests to web servers. When responding to those requests, the web servers use a set of standard HTTP status codes, headers, and, if applicable, payloads.

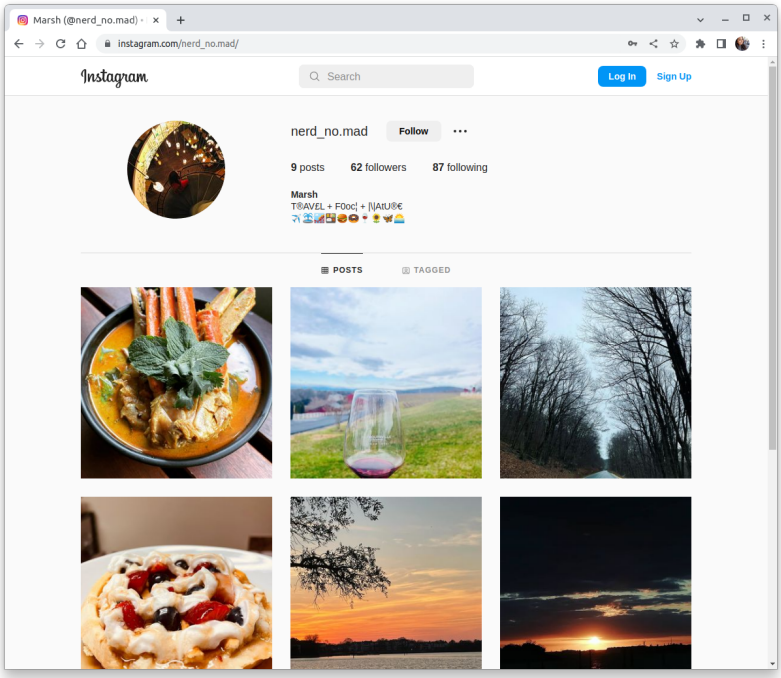

Fig. 7. An Instagram account with the username `@nerd_no.mad` - https://www.instagram.com/nerd_no.mad/

Berners-Lee also authored "Cool URIs don't change" [23] in 1998 which argued that despite the fact that there are countless practical reasons for people to alter their URIs such as website reorganization, document relocation, and a lack of appropriate tools, there are actually no valid theoretical justifications for doing so. He argued that website designers should build their sites so that the URIs remain unchanged for as long as possible.

2.2 STRUCTURE OF SOCIAL MEDIA URIs

The Instagram app was created by Kevin Systrom and Mike Krieger and remained independent until Facebook acquired it in 2012 [24]. We will now examine how URLs are constructed (referred to as the "URL structure" hereafter) on Instagram. The platform decides how its URLs are created, and it is more frequently thought of as both a design decision and a search engine optimization strategy. A carefully constructed URL clearly signals the content at the location to search engines. Figure 7 shows a screenshot of the account page of the user account with the username `@nerd_no.mad`, `https://www.instagram.com/nerd_no.mad/`. Figure 8 shows a screenshot of a post of user `@nerd_no.mad`, `https://www.instagram.com/p/CZQelwuMqT5/`.

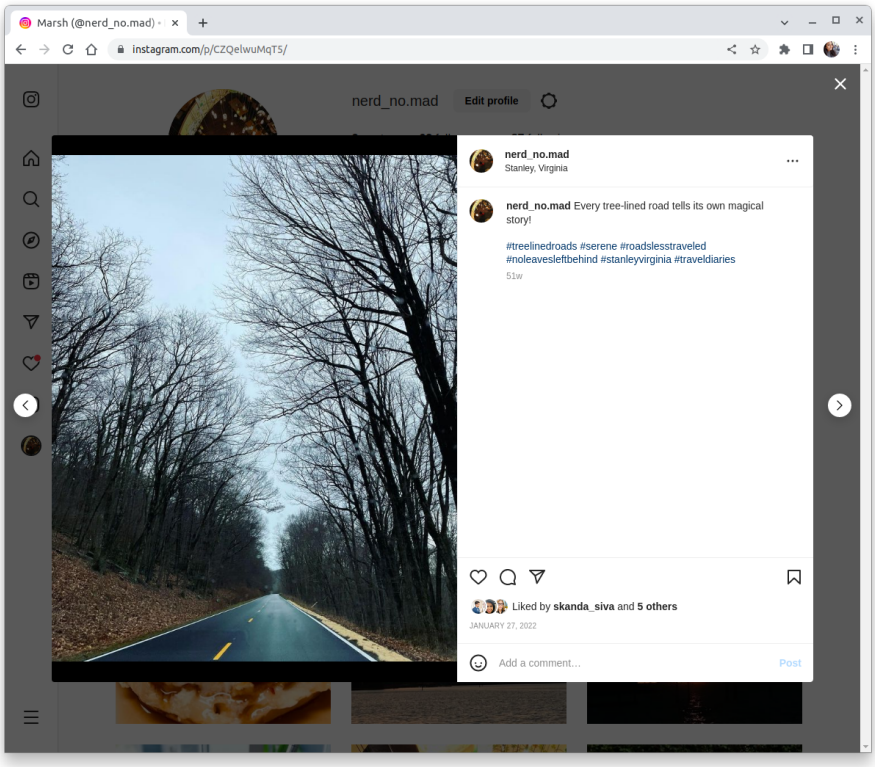

Fig. 8. An Instagram post with the shortcode `CZQelwuMqT5` - https://www.instagram.com/p/CZQelwuMqT5/

The unique code `CZQelwuMqT5` is referred to as "shortcode". This means that the account page and post page URLs both adhere to a general URL structure:

- The Instagram profile of a user has a URL of the form `https://www.instagram.com/{username}`
- The URL of posts that belong to a user follows the format `https://www.instagram.com/p/{shortcode}`

The URL structure of the Instagram post URLs demonstrates how completely opaque they are. By only looking at a post URL we cannot say which user owns this post without dereferencing the URL. This is different from how Facebook (Figure 9) and Twitter (Figure 10) construct their URLs. The URLs for post pages on Facebook and Twitter adhere to a general structure that is built from left to right, as seen below. This shows how the username is present in the URL of a tweet and a Facebook post, but not in the URL of an Instagram post.

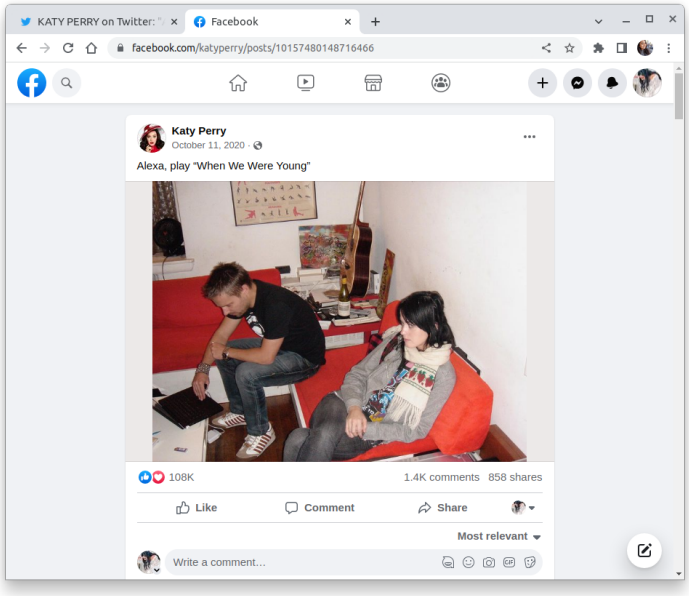

Fig. 9. A Facebook post with the ID "10157480148716466" and username `@katyperry`, `https://www.facebook.com/katyperry/posts/10157480148716466`

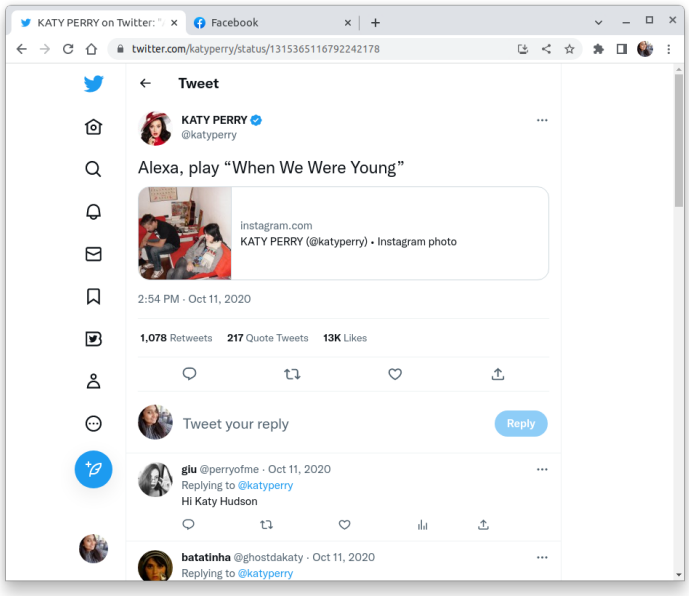

Fig. 10. A Twitter post with the ID "1315365116792242178" and username `@katyperry`, `https://twitter.com/katyperry/status/1315365116792242178`

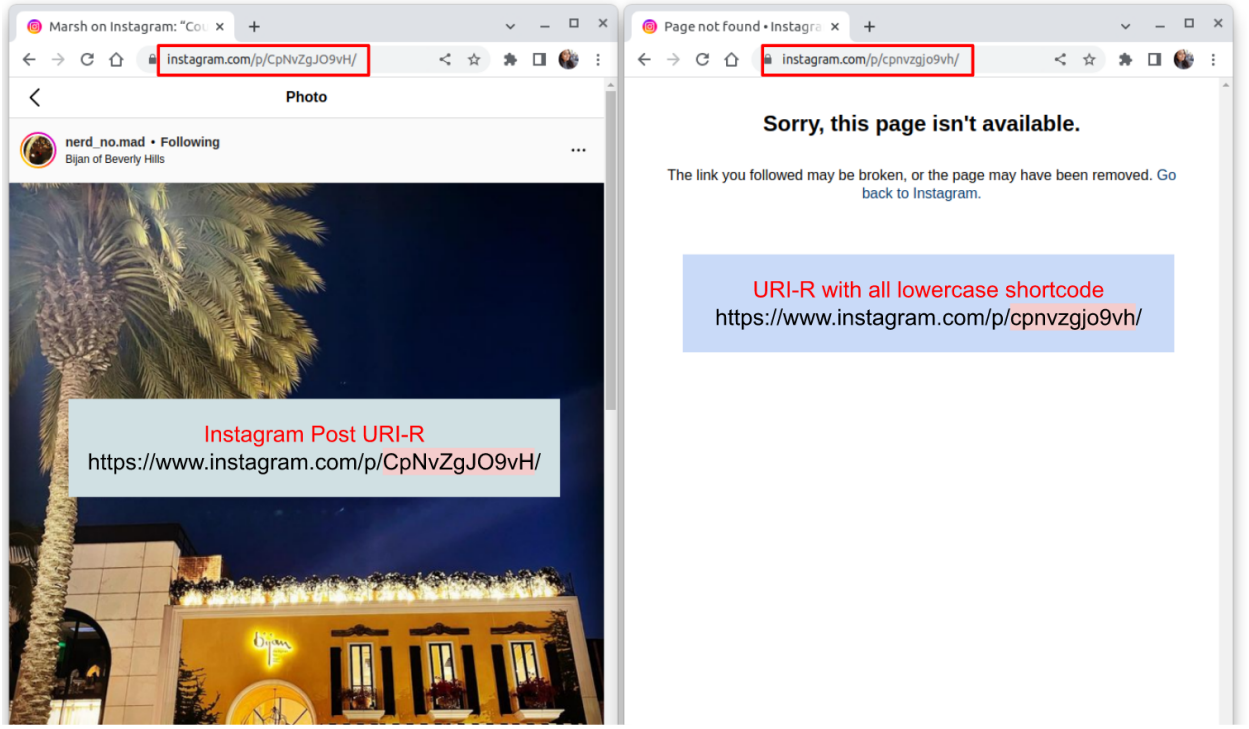

Fig. 11. An example post with the correct post shortcode `CpNvZgJO9vH` (left) and error page for the all lowercased shortcode `cpnvzgjo9vh` (right).
`https://www.instagram.com/p/CpNvZgJO9vH/`

- A post on Twitter (tweet) has a URL of the form:
  `https://twitter.com/{username}/status/{NumericID}`
- A post on Facebook has a URL of the form:
  `https://www.facebook.com/{username}/posts/{NumericID}`

The Instagram shortcode is case-sensitive. For example, the URI-R `https://www.instagram.com/p/CpNvZgJO9vH/` with shortcode `CpNvZgJO9vH` and the URI-R (`https://www.instagram.com/p/cpnvzgjo9vh/`) with its shortcode lower-cased (`cpnvzgjo9vh`) do not lead to the same content. In fact, the URI-R with lower cased shortcode will lead to an error page as shown in Figure 11.

2.3 THE SEMANTIC WEB AND CONTENT-BASED URLS

The Semantic Web, also referred to as Web 3.0, is a development of the World Wide Web with the goal of making Internet data machine-readable. The World Wide Web Consortium's

"Cool URIs for the Semantic Web" [25] outlines standards for creating resource identifiers that prioritize simplicity, stability, and manageability while also providing descriptions that are easily understandable by both people and machines.

Content-based URLs like IPFS [26] and Magnet URIs [27] are used to find files based on their content, not their location. For example, if we are searching for what "IPFS" means, we can visit Wikipedia and enter the term in the search bar, which renders the URL `https://en.wikipedia.org/wiki/InterPlanetary_File_System`. However, if we use IPFS, the URL generated will be `/ipfs/{hash}/wiki/InterPlanetary_File_System`. The hash is the cryptographic hash of the contents in that address and is unique to that content. So instead of asking Wikipedia's servers for the resource, we can ask a multitude of computers in a worldwide peer-to-peer network to share the file, if it is available.

2.4 WEB ARCHIVING

There are services that take the URL of a website as input, for example, web re-hosting services like web proxy, web translator, and web archives [28]. The most popular method for searching for content in web archives is to use the URL of a particular web resource as the lookup key in the search bar. We have highlighted the extant studies that discuss the retrieval techniques to discover mementos in web archives in Section 3.5. As a result of not knowing the URL of the resource, there is a lot of Instagram web content preserved in web archives that the web archive user is unable to discover.

2.4.1 Memento Protocol

An archived web page, or memento (URI-M), is a snapshot of an original resource (URI-R) captured by a web archive at a fixed moment in time (Memento-DateTime). A list of URIs for mementos of the original resource is referred to as the TimeMap (URI-T). Each of the above notations is defined clearly in the Memento Protocol [29]. The Internet Archive's Wayback Machine is one of the most significant and largest web archives that comply with the Memento Protocol.

2.4.2 WARC File Format

The WARC (Web Archive) format [30] is a standard file format developed by the International Internet Preservation Consortium (IIPC)[1] to store web crawls. Concatenating one or more WARC records creates a file in the WARC format. A WARC record is made up of a record header, a record

[1] `https://netpreserve.org/`

content block, and two newlines. The header is required to have named fields for the record's date, type, and length in order to make it easy to retrieve each captured resource (file). Warcinfo, response, resource, request, metadata, revisit, conversion, and continuation are the eight different types of WARC records; we are using only response and revisit records in the implementations in the thesis.

An example WARC response record is shown in Figure 12. The entire protocol response, such as a complete HTTP response with headers and content body from a request, is contained in a WARC response record. It should specify a `WARC-Target-URI` and the `Content-Type` field should contain the value defined by HTTP/1.1, `application/http;msgtype=response`. The header of the response record indicates that it is a unique response to a particular request at a certain time. The header is followed by the content of the original resource, which could be a file or piece of code that can be used to replay the original content in a web browser.

A "revisit" record only contains a compact content block that explains the revisitation of previously archived content. A "revisit" record is often used instead of a "response" record to indicate that the content accessed was either a full or almost a duplicate of previously archived data. Only when it is necessary to reference a previous record in order to understand the current record should a "revisit" record be used; otherwise, alternative record types should be used. An example WARC revisit record is shown in Figure 13. The `WARC-Refers-To-Target-URI` and `WARC-Refers-To-Date` fields, respectively, include the `WARC-Target-URI` and WARC-Date of a record that the current record is a revisit of. This means that if a web archive user accesses a URI-M represented by the `WARC-Target-URI` field (`https://www.instagram.com/designseeds/p/-9KlOQDpvP/`) and `WARC-Date`, the corresponding URI-M that points to the `WARC-Refers-To-Target-URI` (`https://www.instagram.com/p/-9KlOQDpvP/`) and `WARC-Refers-To-Date` would be retrieved.

```
WARC/1.0
WARC-Date: 2019-10-25T02:36:18Z
WARC-Type: response
WARC-Record-ID: <urn:uuid:05772946-e415-4ef6-9043-3358ba583781>
WARC-Target-URI: https://www.instagram.com/p/D/
WARC-Payload-Digest: sha1:4NFVPSJRUOMUF7H2FQGX25XVNFQN5V7I
WARC-Block-Digest: sha1:LEXYO6DUDOETMMZHSPNG6SQVSZASIU22
Content-Type: application/http; msgtype=response
Content-Length: 20715

HTTP/1.0 200 OK
Server: nginx/1.15.8
Date: Fri, 14 May 2021 13:10:58 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Archive-Orig-Vary: Cookie, Accept-Language, Accept-Encoding
...
...
X-Archive-Orig-Content-Length: 55542
X-Archive-Guessed-Content-Type: text/html
X-Archive-Guessed-Charset: utf-8
Memento-Datetime: Fri, 25 Oct 2019 02:36:18 GMT
...
...
b\'<!DOCTYPE html>\n<html lang="en" class="no-js not-logged-in client-root
   ">\n
...
...
```

Fig. 12. An example WARC response record with the `WARC-Target-URI` as `https://www.instagram.com/p/D/`

```
WARC/1.0
WARC-Refers-To: <urn:uuid:96022620-2b12-4677-b7d5-39ba85e59b2e>
WARC-Refers-To-Target-URI: https://www.instagram.com/p/-9KlOQDpvP/
WARC-Refers-To-Date: 2016-09-16T15:27:51Z
WARC-Profile: http://netpreserve.org/warc/1.1/revisit/identical-payload-digest
WARC-Date: 2016-09-16T15:27:51Z
WARC-Type: revisit
WARC-Record-ID: <urn:uuid:e92d61f3-fe6c-4915-ae21-32e86b3e5ab1>
WARC-Target-URI: https://www.instagram.com/designseeds/p/-9KlOQDpvP/
WARC-Block-Digest: sha1:2KOOSJIGG5LTFLNVMCAMEZU4I4XZF4VA
Content-Type: application/http; msgtype=response
Content-Length: 2961

HTTP/1.0 200 OK
Server: nginx/1.19.10
Date: Sat, 02 Oct 2021 16:15:47 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
...
x-archive-orig-content-length: 111181
x-archive-guessed-content-type: text/html
x-archive-guessed-charset: utf-8
memento-datetime: Fri, 16 Sep 2016 15:27:51 GMT
...
Referrer-Policy: no-referrer-when-downgrade
Permissions-Policy: interest-cohort=()
Content-Encoding: gzip
```

Fig. 13. An example WARC revisit record with a `WARC-Target-URI` (`https://www.instagram.com/designseeds/p/-9KlOQDpvP/`) and a `WARC-Refers-To-Target-URI` (`https://www.instagram.com/p/-9KlOQDpvP/`)

```
$ curl -s "http://web.archive.org/cdx/search/cdx?url=https://www.facebook.
   com/katyperry/posts/&matchType=prefix" | head -2
com,facebook)/katyperry/posts/10150201275371466 20160305122514 http://www.
   facebook.com/katyperry/posts/10150201275371466 text/html 302 3
   I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 492
com,facebook)/katyperry/posts/10150295888246466 20111015052123 http://www.
   facebook.com/katyperry/posts/10150295888246466 text/html 200
   TPOUCAFF7PUU34UHMYJCY42KI66HRPSH 13489
```

Fig. 14. A sample CDX prefix query requesting all Facebook posts from @katyperry using the prefix "https://www.facebook.com/katyperry/posts/"

2.4.3 CDX Server API

The CDX is a type of index used in the field of web archiving that provides a simple representation of metadata from all records in an archive. CDX files are created as an index of generated WARC files [31]. This index format contains various fields representing the capture and is often sorted by URL and date. The index that the Wayback Machine uses to lookup captures is served by the standalone HTTP web service known as the Wayback CDX Server API.[2] The CDX server can be used to list every URI-M in the Internet Archive index for a given URI-R. We can use the CDX API to return results matching a specific "prefix", even though the default response of the CDX API is set to "exact" match.

For Facebook, a CDX prefix query http://web.archive.org/cdx/?url=facebook.com/{username}/posts/&matchType=prefix would provide the web archive user with all the available Facebook status posts that belong to a particular user. For example, Figure 14 shows a query to the CDX API, requesting for post URI-Ms (with the term "posts" in the URI) that belong to the user account, @katyperry on Facebook.

Similarly for Twitter, a simple CDX prefix query http://web.archive.org/cdx/?url=twitter.com/{username}/status/&matchType=prefix would help a web archive user to identify all the available Twitter status posts that belong to a particular user. For example, Figure 15 shows a query to the CDX API, requesting for post URI-Ms (with the term "status" in the URI) that belong to @katyperry user account on Twitter.

[2] https://github.com/internetarchive/wayback/tree/master/wayback-cdx-server/

```
$ curl -s "http://web.archive.org/cdx/search/cdx?url=https://twitter.com/
   katyperry/status/&matchType=prefix" | head -2000 | tail -2
com,twitter)/katyperry/status/1068608473913409536 20211206195744 https://
   twitter.com/katyperry/status/1068608473913409536 text/html 200
   BA4YBDRPGCEYUWF3KZHHXSUTZ6H5AH7T 7146
com,twitter)/katyperry/status/1068608473913409536 20211223104256 https://
   twitter.com/katyperry/status/1068608473913409536 text/html 200
   ILRY7X65W6Z64YSYHN7NQSWCGDJ6F2GI 7171
```

Fig. 15. A sample CDX prefix query requesting all Twitter posts from `@katyperry` using the prefix "https://twitter.com/katyperry/status/"

2.4.4 Client-Side Rewriting

Client-side rewriting is the process that alters a web page's content before it is displayed on the client-side (i.e., in the user's web browser itself). In this instance, scripts are used to intercept the HTML and CSS that are being sent from the server to the client and to make any necessary modifications before the website is rendered. This is achieved using JavaScript or other scripting languages to manipulate the Document Object Model (DOM) of the web page, which represents the structure and content of the page. The rewrite of URLs on the archived page to refer to the archived versions of the associated pages rather than the versions on the live web is one common use case for client-side rewriting in web archives. This makes sure the user can move around the archived web easily and consistently. Wombat[3] is a JavaScript library which is a standalone client-side URL rewriting system that performs the rewrites through targeted JavaScript overrides [32]. In Section 4.3.2, we propose another JavaScript in addition to "wombat.js" to help us achieve our goal.

2.5 TRANSIENT NATURE OF SOCIAL MEDIA ACCOUNTS

We note that as of January 2023, Twitter has restored Naomi Wolf's account after Elon Musk acquired the company and began restoring many suspended accounts [33]. Despite the fact that her Twitter account is currently active, the examples used in Section 1 hold without loss of generality. We are aware that social media platforms often ban accounts for various reasons. For example,

[3]https://github.com/webrecorder/wombat

Kanye West's Instagram account was restricted in early October 2022 as a result of an anti-Semitic post by him [34]. At the end of October, he was allowed to post again on Instagram, but after four days, he was immediately banned for making another anti-Semitic post [35]. His account was briefly reinstated by Instagram again before being banned again for the third time within a couple of months for sharing a clip of the song "Someday We'll All Be Free" [36]. As of January 26, 2023, his Instagram account is currently active after being reinstated in mid-January 2023. Another recent example is how Twitter suspended the account of white supremacist Nick Fuentes on January 25, 2023, a day after its restoration [37]. He was initially banned from Twitter in July 2021 and was reinstated by Elon Musk along with other accounts of many people who had been banned for advancing far-right extremist ideas on the platform. By using Kanye West's and Nick Fuentes's social media accounts as examples, it is evident that account restoration is also not permanent; an account may be suspended or restored today, but there is no way of knowing what will happen tomorrow.

CHAPTER 3

RELATED WORK

This chapter provides a review of the existing literature on social media and web archiving.

3.1 THE IMPACT OF SOCIAL MEDIA

With the growing increase in social media popularity, we have seen a rise in social media research recently. There is a wide range of social media research currently available, covering topics such as the impact of social media on mental health [38, 39, 40], political outcomes [41, 42, 43], and security and privacy [44, 45, 46, 47]. In-depth research has also been conducted on social media marketing and business research [48, 49, 50, 51, 52] as well as the dissemination of misinformation and disinformation on social media [53, 54, 55, 56, 57], particularly in the wake of the COVID-19 global pandemic. These studies show that social media has an impact on various aspects of society. It has revolutionized the way people communicate and share information. Understanding these interactions on social media sites can help us better understand social dynamics and human behavior.

A social media Application Programming Interface (API) allows interested users (developers, marketers, researchers, etc.) to access and retrieve data from social media platforms. The data that can be accessed through social media APIs usually includes user profiles, posts, comments, likes, shares, and other interaction metrics. In one of the earliest studies, Lomborg et al. [58] discussed the benefits and challenges of using APIs to collect data from social media platforms. The authors also emphasize how crucial it is to abide by platform terms of service and data protection laws while using social media APIs. However, it is important to know that APIs are constantly evolving, the functionalities available today may not be the same in the future and changes to API access policies can happen anytime. This means that researchers need to keep up with API changes and updates to ensure that their data collection methods are up-to-date and accurate. In 2017, Hogan [59] examined the impact of Facebook's API changes on users' social networks. The paper discusses how disruptions and the loss of some features in the app caused by changes to Facebook's APIs have affected users' capacity to manage their social networks. In 2019, Bruns [60] investigated the difficulties social media researchers faced in getting data from social media platforms because of modifications to platform policies and restrictions on API. The author also emphasizes the necessity of addressing these challenges, including the potential adoption of alternative data collection techniques such as web scraping, despite its questionable legal status. The author makes

the case that these challenges are a sign of a wider trend of data commercialization and privatization, which has consequences for academic research. A study conducted by Hartman et al. [61] in 2021 explored whether modifying the terms and conditions of the Instagram API, such as restricting third-party applications from accessing data, would impact the quantity of research conducted on Instagram. Their research found that, despite the 2018 API changes, there was no significant shift in research topics or decrease in the number of published articles; rather, the trend continued to increase.

Compared to the other social media platforms, there is a concern that Twitter research may be over-represented in academic studies due to its ease of access for researchers, as it provides open access to its data through its API (although Twitter has ceased free access to its API in February 2023 [62]). Recent academic research on Twitter has focused on a variety of topics, including sentiment analysis [63, 64, 65], political communications [66, 67, 68], and computational methods like tools and algorithms for analyzing large datasets [69, 70]. Additionally, there have been studies about conflict situations like the current war between Ukraine and Russia [71, 72, 73]. There have also been other studies that have even examined Twitter research in the context of its ability to obtain real-time large-scale datasets, data quality, data sampling, and not to mention the reliability of the Twitter API [74, 75]. On the other hand, platforms like Facebook and Instagram have more stringent data access policies, which makes it more challenging for researchers to study these platforms. As a result, there may be a bias in the literature toward Twitter research, which may not accurately reflect the broader social media landscape. In our study, we are focusing on the Instagram platform. The goal of our study is to support researchers interested in studying Instagram posts through web archives to easily access mementos of all the posts of a particular user on Instagram. This focus on Instagram will enable a broader and more detailed comprehension of social media, beyond just the Twitter platform.

Instagram has shown huge potential as a platform for news reporting. There are studies that indicate how Instagram has developed into a significant platform for news distribution and journalism, including ones that looked at how Instagram Stories are used to disseminate news [76], and how the BBC used Instagram posts to establish a narrative for reporting on Afghanistan [77]. Al-Rawi et al. [78] also compared which news stories Instagram audiences most frequently "liked" or "reported". These studies demonstrate how journalists and news organizations can use Instagram to reach larger audiences and engage with them in novel and innovative ways, but they also highlight the difficulties and limitations that come with the platform due to its emphasis on visual content and its transient nature.

Instagram has evolved from a simple photo-sharing app to a full-fledged marketing platform

over the years, with organizations of all sizes and kinds using the platform to promote their work. Hu et al. [79] conducted one of the earliest studies on Instagram by looking into its users and their behavior. The authors used Instagram posts to explore the different types of user-generated content (selfies, pictures with friends, pictures of food, etc.), different types of users based on the content they post (for example, selfie-lovers who mostly post selfies or common users who post all the categories of photos), and how the type of the user and the content they post affect the users' number of followers. The authors learned that the number of followers of a user was independent of the type of user they are. There have also been studies that suggest how Instagram can be used by different institutions to engage with their audiences. For example, Doney et al. [80] conducted a content analysis of academic library Instagram posts which revealed that more than 50% of shared posts featured events, services, or resources from libraries or universities. They also found that the number of likes differed based on the different categories of posts. Another example of how institutions use Instagram to enhance their online presence is explained in a case study conducted by Wilkinson [81]. She described how the Mansfield Library Archives and Special Collections embedded the most recent posts from their Instagram feed to add timely and interesting dynamic content to their otherwise static website. In late 2016, Bamber [82] discussed her experience in using Instagram to promote web archive collections. She emphasized that Instagram is not only cost-free and easy to use, but it also allows you to connect with a bigger audience. Over the span of six months, their Instagram account's followers grew from 40 to over 3,000, but only by 200 on Facebook.

3.2 THE IMPORTANCE OF WEB ARCHIVING SOCIAL MEDIA

Despite the numerous studies conducted on social media such as the ones we discussed in Section 3.1, those platforms remain as "black box" services with opaque inner workings that are not transparent to users. The user interacts with the service through a User Interface (UI), but the underlying mechanisms that power the service are hidden from the user's view. As researchers, we are only able to examine the versions of these services on the live web to conduct studies. However, people often delete their social media posts and sometimes their entire accounts. Some even have their accounts temporarily or permanently suspended by the platform's moderators. Elon Musk's recent acquisition of Twitter serves as an indication of how administrative changes can alter any platform, including significant adjustments to platform features and the suspension and reinstatement of user accounts [83]. Additionally, these platforms may merge together or even cease to exist over time, making it impossible to guarantee their availability for users in the future. For example, Friendster and Google+ are two examples of social media platforms that are

no longer in operation. Friendster was an early social media platform that launched in March 2002 but suspended its services in June 2015 due to the lack of engagement, site performance issues, and competition by other platforms [84]. Similarly, Google+ was a social network that launched in June 2011 and was shut down in April 2019 due to low usage and challenges with keeping the platform secure [85]. Finally, the UI changes that social media platforms frequently undergo can significantly affect how users interact with the platforms and the content that is made available to them. Preserving social media can help avoid the loss of crucial data and allow us to study historical end-user web experiences, access content that might not be available on the live web, and conduct research on subjects like user behavior, trends, and the evolution of web communications. Overall, the importance of preserving social media content cannot be overstated, as it provides valuable insights and information that can be used for a variety of purposes, both now and in the future.

Researchers have made attempts to understand and estimate the size and complexity of the web. Xing and Paris [86] proposed a method for accurately measuring the size of the web using importance sampling. The authors calculated the size of the Internet Protocol Version 4 (IPv4) Internet by counting the number of publicly accessible web servers and FTP servers, which they claim to be accurate estimates. The WorldWideWebSize.com [87] estimates the size of the World Wide Web every day based on the estimations of the number of pages indexed by top search engines. Over the years, researchers have raised apprehensions about how easily resources on the web can disappear or change and emphasized the significance of preserving the web. In 2009, Parks [88] expressed his concerns about the potential loss of valuable information that could result from the disappearance of Internet resources. Parks emphasized the significance of preserving web content and developing long-term systems for managing it, while also acknowledging the difficulties that future researchers might encounter in the absence of digital resources.

With ongoing efforts to assess the vastness and intricacy of the web, concerns about preserving web content for future generations have been raised. In 2011, Ainsworth et al. [89] investigated "How much of the web is archived?" The study investigated various methodologies and challenges involved in gauging the extent and completeness of web archives, as well as approaches to assess them. The researchers used a subset of URIs to approximate the proportion of archived URIs, as well as the count and frequency of archived versions, which they extrapolated to infer the percentage of the surface web that is encompassed by web archives. Milligan [90] in 2016 pointed out that the sheer volume of web pages can be both an advantage and a disadvantage. According to him, future historians will have to face the challenge of finding information that was significant at the time to different communities. He observed that there was a shortage of efficient tools to help historians navigate web archives, and the few that did exist at the time were still in their initial

developmental stages.

It is difficult to determine the exact proportion of the Internet that is comprised of social media because this is a constantly evolving setting with new platforms appearing and existing ones disappearing. However, it is estimated that social media accounts for around 90% (4.76 billion social media users out of 5.16 billion Internet users) of Internet usage and activity worldwide as of January 2023.[1] While social media platforms may represent only a portion of the Internet, even a small percentage of a colossal entity remains significant. Accordingly, our research is geared towards making it easy for web archive users to discover mementos of Instagram posts that pertain to a specific user.

3.3 CAPTURING THE EVER-EVOLVING: EXISTING STUDIES AND CHALLENGES OF WEB ARCHIVING SOCIAL MEDIA

It is important to point out that web archiving should not be confused with social media archiving. In 2017, Littman [91] examined the differences between the two and came to the conclusion that those two forms of archiving should be regarded as complementary. He points out how web archiving focuses on capturing and preserving entire websites and all of their components, including images and videos, and replaying them for users whereas social media archiving typically involves the gathering of social media data through APIs. While both of them involve the preservation of digital content, we can favor one over the other depending on our requirements. According to Littman, if our goal is to experience the platform (the look and feel), accessing social media through web archives is better. If our interest is in social media as data, however, the API collected social media will be very beneficial. Subsequently, Littman et al. [92] conducted a comprehensive analysis of the practical elements of implementing API-based social media data collection as a method of web archiving. The authors discuss Social Feed Manager (SFM),[2] a web-based tool designed to support web archiving initiatives by helping collect data from different social media APIs.

Studies on web archiving are many and diverse, and it has emerged as a crucial method for preserving digital material. The creation of infrastructure and services that can handle large collections is an important factor in web archiving. In one of the most recent studies, Wang suggested the development of a web archive infrastructure that can facilitate the efficient and scalable web archiving services that enable better quantitative data analysis and browsing, using big data formats like Parquet instead of WARC [93]. Many researchers have turned to social media collections to study the COVID-19 pandemic's effects because it has had such a global influence. Individuals

[1] https://www.statista.com/statistics/617136/digital-population-worldwide/

[2] https://www.github.com/gwu-libraries/sfm-ui

or institutions can create and preserve their collections using Archive-It,[3] which is a subscription service provided by the Internet Archive. The National Library of Medicine's (NLM) Web Collecting and Archiving Working Group[4] has created the "Global Health Events web archive"[5] using Archive-It to preserve web-based content about the COVID-19 pandemic. Speaker et al. [94] provide details on the scope of the project, the types of content collected from various sources such as websites, blogs, and social media, the methods and criteria used for collection, and other groups' efforts to collect COVID-19 related content. The "COVID-19 Web Archive"[6] is an initiative that brings together more than 154 individual web archive collections from over 120 libraries, archives, and cultural heritage organizations, including NLM's collection. In a series of blog posts, Thomson [95, 96] discussed the potential of web archives as a means of preserving social media content in addition to the official websites related to the pandemic. She also discussed the challenges of archiving social media content such as the sheer volume, the rapid pace of content creation, and legal and ethical considerations. Despite these challenges, Thomson argued that social media platforms offer valuable insights into public responses to the pandemic and its impact on the general public.

In the realm of social media, tools and frameworks have been developed to aid in web archiving, such as yourTwapperKeeper [97] and the Profile-Based Focused Crawler for Social Media-Sharing Websites [98]. They enabled the collection of social media content and can be used for various research purposes. It is important to note that not every web archive capture is absolutely perfect. While web archives are an effective tool for preserving web content, there are potential issues that can impact their quality and usefulness, particularly with dynamic content such as social media content.

A significant challenge associated with social media captures is the absence of resources such as images, videos, or even entire web pages during playback. The cause of this problem could stem from technical limitations, errors made during the archiving process, or changes made to the original website, which can prevent access to specific content. Garg et al. [11, 99, 100] investigated the issues with replaying archived Twitter content in 2021. Because of changes to the Twitter platform (UI change in June 2020) and the dynamic nature of social media, archived new UI Twitter pages were quite often incomplete or inaccurate. Researchers have also attempted to measure the quality of an archived web page. For example, Memento Damage [101, 102] is an open-source tool that detects inconsistencies and inaccuracies in archived web content. The Archival Acid Test, on

[3] https://archive-it.org/

[4] https://www.nlm.nih.gov/webcollecting/index.html

[5] `https://archive-it.org/collections/4887`

[6] `https://covid19.archive-it.org/`

the other hand, is a framework to evaluate archive performance on advanced HTML and JavaScript since ensuring the quality of archived content is crucial [103].

"Link rot" is a term used to describe any situation where web links become outdated or broken over time, often as a result of changes to the web pages they point to [104]. It is a broader concept than reference rot, which specifically refers to links in citations or references becoming inaccessible or changing over time in scholarly communication. SalahEldeen and Nelson [105] investigated the prevalence of link rot in social media by using a large dataset of web links extracted from tweets and other social media content. They discovered that after a year, around 11% of the links had become inaccessible and only 20% had been archived. Researchers have suggested potential solutions such as using persistent identifiers and giving each hyperlink its own metadata record that contains pertinent data about the linked content [106, 107, 108, 109] in order to prevent reference rot and to increase reliability. Content drift is the phenomenon where the content of a web page changes over time, frequently leading to discrepancies between different versions of the page [110, 106, 104]. An instance of content drift can be observed when a website originally focused on a specific topic is modified with content related to a different topic. Jones et al. [110] looked into content drift specific to URI references in scholarly articles and they found out that 75% of these resources has changed from what it was at the time of publication, highlighting the need for techniques to combat these problems.

Although it is crucial to address the technical challenges that arise in this area of web archiving, it is also necessary to take into account the legal and ethical issues. Sharma [111] notes that legal issues in web archiving have been thoroughly discussed in scholarly discourse, but the same cannot be said for the emerging field of social media archiving. She specifically examined Twitter as a case study and identifies three major legal issues that libraries and archives might face while building a Twitter archive: copyright, privacy, and right of publicity. The thesis by O'Halloran [112] focuses on the challenges of archiving controversial social media content, specifically using the example of the Irish slave's meme. Despite this false and hateful content, the paper argues that it should be preserved as a snapshot of race relations and the dissemination of racist ideas in the modern era. The central focus of the thesis is the role that archivists can play in preserving problematic and false information spread through social media and the issues that an archival institution must address when preserving such content. Despite these technical or ethical challenges, web archives remain a valuable resource for researchers and historians studying the evolution of the Internet and its impact on society.

3.4 EXISTING STUDIES ON WEB ARCHIVING INSTAGRAM

Web archiving Instagram entails capturing digital content from the platform to prevent content loss and to enable researchers to examine the emergence of Instagram as a social media platform and its impact on society over time. Duncan et al. [113] discussed the web archiving practices used by the New York Art Resources Consortium (NYARC) for archiving born-digital ephemeral art resources. The authors talked about how the creation and upkeep of web archives can be broken down into a four-stage lifecycle: collection development and curation, harvesting and quality assurance, storage and preservation, and description and access. This life cycle is applicable to all programs that gather, preserve, and make resources accessible to the users. Bainotti et al. [114] discussed the popularity of ephemeral content with the use of Instagram Stories (which only last for 24 hours) and challenged the traditional web archiving approaches while highlighting the need for new approaches to preserve ephemeral content.

In 2017, MacDowall et al. [115] examined the impact of Instagram on the way street art and graffiti are produced, consumed, and perceived by audiences. MacDowall et al. draws attention to the fact that using Instagram only as a data source ignores ethical and legal issues like privacy concerns, archiving, and conservation that comes up when undertaking research with digital material. The IIPC's Content Development Working Group[7] has recently taken up the challenge of web archiving content related to street art using image databases, blogs, and news but social media sites like Instagram are considered out of scope for their collections since it is labor intensive and highly unlikely to be archived successfully [116].

Numerous web archiving initiatives have been started to attempt in preserving ephemeral content. The Stanford Libraries' web archiving program brought up the fact that they use Archive-It as the preferred solution for archiving topical web archive collections but there are sites like Instagram which are not able to be archived successfully by Archive-It [117]. Despite the fact that some of these sites can be captured with services like Conifer,[8] they note that their current web archiving workflow is incompatible with files produced by those services. Using the Conifer toolkit, they developed a brand-new workflow to capture Instagram content, which will be stored in the Stanford Digital Repository (SDR) and made available through the ArchiveWeb.Page[9] interface.

Earlier we listed some studies related to the spread of misinformation and disinformation on social media broadly and it is now pertinent to direct our attention toward Instagram in particular. The Center for Countering Digital Hate (CCDH)[10] disclosed evidence that Instagram's algorithm promotes COVID-19 and vaccine misinformation [118]. Moreover, they discovered that the posts

[7] https://netpreserve.org/about-us/working-groups/content-development-working-group/

[8] https://conifer.rhizome.org

[9] https://archiveweb.page

[10] https://counterhate.com

related to vaccines were lacking the intended information labels and the algorithm failed to identify those posts. A subsequent report by the CCDH [119] has identified the Disinformation Dozen, a group of twelve individuals, as the most prominent spreaders of COVID-19 disinformation on social media platforms. Bragg et al. used the Instagram account pages of both the Disinformation Dozen and certain health authorities to gauge the replayability and quality of Instagram account page mementos at the Wayback Machine [14, 15]. The study found that only about 1% of mementos for the Disinformation Dozen's Instagram accounts are replayable with complete post images. The authors discovered a challenge in archiving Instagram because non-logged-in users are eventually redirected to the Instagram login page. As per the study, 96.13% of mementos from the Disinformation Dozen accounts redirected to the login page. This means that the number of mementos listed for these users' account pages Wayback Machine interface can be misleading, as most mementos are actually redirections to the Instagram login page.

3.5 EXISTING APPROACHES FOR RETRIEVING ARCHIVED CONTENT FROM WEB ARCHIVES

Ensuring the ease of retrieving archived social media content is just as critical as the archiving process itself. Accessing and analyzing archived social media content easily is essential for researchers and historians studying the evolution of the Internet through web archives. Therefore, it is imperative to incorporate retrieval methods and technologies into the archiving process to guarantee the archived content's usability and accessibility.

In 2009, Jaffe and Kirkpatrick [120] discussed the architecture of the Internet Archive and how it is an essential component of efficient web archive searching. The authors pointed out that the system requires a search and index mechanism to enable users to locate specific items. They further discussed the implementation of each of the processes, including storage, indexing, and searching, and highlighted the technical challenges associated with web archiving. In one of the earlier studies, Michael Stack introduced the open-source full-text search of web archive collections developed by the Internet Archive in partnership with the IIPC. In his work, Stack continues to discuss the difficulties associated with searching web archive collections and explains how these issues were overcome by using an open-source search engine known as Nutch [121, 122]. In addition to the Internet Archive, researchers have extensively studied search strategies used by other web archives such as the Portuguese Web Archive, Archive-It, and National Library of the Netherlands [123, 124, 125].

A survey in 2011 by Daniel Gomes et al. [126] reported that 89% of web archives support URL-based search, which has the drawback of requiring users to remember specific URLs while 79% of web archives provide metadata search based on category or theme. According to the same survey,

67% of web archives allow full-text search, the most popular method of information access used by web search engines despite the significant computational demands. In 2013, Ahmed AlSum et al. [127] addressed how using the original URI-R as the lookup key in web archives (URL-based search) is simple until the resource in the live web issues a redirect ($R \rightarrow \grave{R}$). It is difficult to know prior to the search which of these two URIs (R or $\grave{R}$) should be used to discover archived copies of the resource. The authors introduced new guidelines that will assist the client in using HTTP redirection to obtain a closer Memento to the requested date time. Costa et al. [128] brought up several important discussion points that affect the efficiency of web archive search and scalability. The authors mention that the URL-based search is the prevalent access method in web archives due to the challenge of indexing all the collected data. The authors discussed a variety of problems that limit the efficient web archive search, such as the information overload brought on as web archives are required to accumulate all previous documents and indexes unlike tossing the outdated versions out when new ones are found, as search engines do and the unstructured nature of data. They also emphasized that even if it were technically feasible, building a central index for all the data in online archives would be ineffective.

Ming et al. address various retrieval techniques and technologies that can be used to streamline the search in the context of web archiving [129]. In their discussion, they point out that web archives previously relied solely on using URLs as keys for searching through archived web pages, which was not the most efficient method. However, they note that in recent years, as web archives have expanded in size, more effective search methods have been developed and implemented. In their research, they investigated the impact of using appropriate hashing algorithms locally to enhance the speed of web archive searching. They also demonstrated how different hashing methods can be employed to shorten and organize URLs, thereby improving the efficiency of the search process.

A couple of studies explored different methods for searching web archives without relying solely on traditional indexing. For example, by focusing on searching the Internet Archive specifically, Kanhabua et al. proposed an entity-oriented search system to enable analytics and retrieval in the Wayback Machine [130]. In order to enable keyword searches without actually processing and indexing the raw archived content, they used the Bing search engine to obtain a rank list of results from the live web and linked those results to the Wayback Machine. Vo et al. [131] studied ranking techniques that make use of metadata rather than the complete content of documents in order to facilitate web archive search. They recommended a learning model that combines different types of metadata to differentiate between the relevancy of search results, and they carried out tests to verify its efficacy. Their research was a first step toward using metadata to improve website

ranking in web archives.

By investigating how other researchers have used the CDX API, we were able to identify both its applications and limitations. Numerous studies use the CDX API to collect data because it is easier to access and analyze than getting the mementos themselves. Siddique [132] was the first to enumerate all of the URL variants available for a Twitter account page using the CDX API with a URL match scope of "prefix". The same technique was used by Summers to collect tweets from former president Donald J. Trump's personal Twitter account (`@realDonaldTrump`) at the Internet Archive [133]. Our investigation helped us recognize that we could support more studies similar to those carried out on Twitter, but on the Instagram platform, by providing "prefix" search capabilities for Instagram posts.

There exist several other initiatives that have been developed either by using CDX files, CDX server API output, or simply with the aim to improve the performance of accessing web archives. In order to create a research corpus using distributed web archive processing, Holzmann et al. [134] developed a framework called ArchiveSpark that makes use of standardized data formats like the output from the CDX index of web archives. ArchiveSpark operates more effectively, leading to quicker processing times, by choosing records of interest from the CDX index before retrieving the original archived content from the disk. The authors used Warcbase,[11] as the baseline in their benchmarking process. Kelly et al. [135] made use of the CDX server API output containing metadata such as the HTTP status code of the capture and Sort-friendly URL Reordering Transformed (SURT)[12] URIs in their research on how URI canonicalization affects the number of mementos discovered for a particular resource. In their study, the authors used MemGator's [136] output in the CDXJ format, an extension of CDX to enable fast and accurate parsing of DateTime information associated with the mementos listed in multiple web archives.

Alam et al. [137] used CDX files from the UK Web Archive, Stanford University Archive, and the ODU dark archive of Archive-It to investigate web archive profiling through CDX summarization used to guide the routing of the requests in the Memento aggregator. They examined the trade-offs between routing efficiency and profile size and discovered that they could achieve a 22% routing precision (accurately predicting that the requested URI is present in the archive) without encountering any false negatives. They noted that this will allow the Memento aggregator to considerably save time and resources by only querying the archives that are most likely to have a copy of the desired URI. Maurer [138] proposed cdx-summarize,[13] a toolset to generate summary files for web archive collections, which researchers can use to explore and compare collections from

[11] https://github.com/lintool/warcbase

[12] `http://crawler.archive.org/articles/user_manual/glossary.html#:~:text=SURT`

[13] `https://github.com/ymaurer/cdx-summarize`

different institutions without facing legal restrictions or unwieldy amounts of data. He showcased how researchers could apply the output of this tool by illustrating how they could generate compelling summaries about the Luxembourg Web Archive's content. Furthermore, he demonstrated how researchers can use this tool to compare archives by leveraging the Luxembourg Web Archive, Common Crawl,[14] and the Internet Archive's CDX server. Alam and Graham [139] most recently developed a tool called CDX Summary,[15] an open-source tool that can create statistical reports based on various attributes such as URIs, hosts, TLDs, status codes, etc. This tool uses the CDX index from a collection of WARC files at the Internet Archive.

3.6 SUMMARY

In this chapter, we discussed the existing literature to acquire insights into how social media impacts society, the significance of web archiving social media, and the challenges that arise when web archiving social media platforms, with a particular emphasis on Instagram. Furthermore, we explored different studies on retrieving archived content from web archives to identify the existing approaches.

The available literature on social media highlights its significant impact on society, making it necessary to archive social media content. While researchers could use social media APIs to access data such as user profiles, posts, comments, likes, and shares, they need to comply with platform terms of service and stay up-to-date with API changes. The existing work showed that Twitter is often studied because it provides open access to its API, but other platforms like Facebook and Instagram have strict data access policies. We realized that social media has been extensively researched for its impact on mental health, politics, security, privacy, business, news, and journalism. Unfortunately, the spread of fake news and misinformation is also a concerning issue that has been studied in relation to social media.

Our observations of social media revealed that it provides valuable insights and information for researching user behavior and trends. This underlines the importance of web archiving social media, including Instagram, as these platforms are significant in capturing current important societal events. We recognized the crucial role of preserving social media content for historical purposes and research, as these platforms undergo constant change and evolution and even disappear over time, making it difficult to retrieve important information that was once available.

The existing studies on web archiving social media have documented challenges related to web archiving social media and they reveal a pressing need to enhance web archiving techniques, par-

[14] `https://index.commoncrawl.org`

[15] `https://github.com/internetarchive/cdx-summary`

ticularly for social media platforms. The studies have shed light on several tools and frameworks designed to tackle some of these issues like link rot, content drift, scalability of managing large web archives, and problems during replay like inconsistencies and missing resources in archived web content. The studies demonstrated the keenness of researchers and developers to enhance web archiving processes.

Through exploring existing approaches for retrieving archived content from web archives, it becomes apparent that the ease of access to archived social media content is as important as the archiving process itself for researchers and historians studying the evolution of the Internet through web archives. We discussed how the CDX API has been used by different projects to collect data for analysis, specifically for Twitter and Facebook using a prefix search. To the best of our knowledge, no study has attempted to discuss approaches to retrieve all archived Instagram posts from a specific user. However, the earlier research has provided valuable insights into the necessity of having such a feature for web archive users.

Thus, the knowledge acquired from the existing studies collectively emphasizes the importance of enabling easy access to archived Instagram posts for users who possess an interest in such content. This is especially relevant for current and future historians or researchers who use web archives for their scholarly pursuits. Our work not only highlights the need for further research in this area but also encourages more studies where archived Instagram posts could serve as a valuable primary source.

CHAPTER 4

APPROACHES AND EXPLORATORY EVALUATION

In Chapter 1, we outlined the challenges associated with retrieving archived Instagram posts that belong to a particular user from web archives. Although it is possible to query for all posts belonging to a specific user in web archives for platforms like Facebook or Twitter using a CDX prefix query, such a direct approach is currently unavailable for Instagram posts. This is mainly due to the URL structure of Instagram, which lacks the username of the post owner in the post URL. In this chapter, we will present our proposed solutions for addressing this issue, along with an exploratory evaluation of their implementation feasibility.

In this chapter, we will discuss two primary approaches for facilitating the retrieval of all archived Instagram posts attributed to a specific user from web archives:

1. Using WARC revisit records to incorporate Instagram usernames into the `WARC-Target-URI` WARC record header (Section 4.2).

2. Building a secondary index to map user accounts to their post URLs (Section 4.3).

4.1 DATASETS

This study employs multiple datasets as summarized in Table 1. The IAlogs and IAholdings datasets are used to demonstrate our approaches, whereas the KatyPerryLive and BlakeLivelyArchive are created as a result of our demonstrations.

Table 1. An overview of the datasets used.

Dataset Name	Data	Description
IAlogs	2,435,718 requests	Instagram access logs sample from the Internet Archive.
IAholdings	280,292 URI-Rs	Instagram URI-R sample from the Internet Archive holdings.
KatyPerryLive	1642 posts	Posts extracted from Katy Perry's account page from the live web.
BlakeLivelyArchive	87 posts	Posts extracted from Blake Lively's account page from using the mementos of her account page at the Internet Archive.

Table 2. Dates used from the web archive server access logs. These datasets from the first Thursday of February each year (2011-2021) are combined to create the IAlogs dataset.

Year	Date	No. of Requests (Total)	No. of Requests (Instagram)	Percentage (%)
2011	2011-02-03	43,427,724	194	< 0.01
2012	2012-02-02	99,173,542	1,143	< 0.01
2013	2013-02-07	129,751,846	5,094	< 0.01
2014	2014-02-06	108,818,918	13,275	0.01
2015	2015-02-05	143,517,276	29,222	0.02
2016	2016-02-04	170,903,561	127,818	0.08
2017	2017-02-02	179,471,610	161,382	0.09
2018	2018-02-01	276,532,759	457,498	0.16
2019	2019-02-07	308,194,920	544,253	0.18
2020	2020-02-06	498,393,851	637,442	0.13
2021	2021-02-04	386,928,532	458,397	0.12

The IAlogs dataset used for our research includes web archive server access logs from the Internet Archive [140]. We used access logs that cover one day per year from 2011 to 2021, and specifically filtered out the logs related to Instagram. Table 2 shows the specific dates from each year, the total number of raw requests, and the number and percentage of Instagram requests filtered from the total number of requests. The date of each dataset is the first Thursday in February, which represents an average day of the year. There is a rise in the number of requests for the Instagram domain over the years, particularly up until 2020, but it is noteworthy that the proportion of Instagram requests remains consistently less than 0.2%. Table 3 shows the top 20 most requested posts from these logs along with their manually extracted usernames, which we have used in various demonstrations that will be discussed in Sections 4.2 and 4.3.2.

The IAholdings dataset is a sample of the Internet Archive holdings. We acquired this sample of URI-Rs through a separate research project that aimed to revisit the question of how long a web page lasts [141]. Their sample consists of 285 million unique URI-Rs, from which our subset includes 280,292 URI-Rs that belong to the domains `instagr.am` (4324 URI-Rs) and `instagram.com` (275,968 URI-Rs). Instagram had used `instagr.am` as one of their earlier domains, and it now redirects to `instagram.com` as shown in Figure 16. Our sample of Instagram URI-Rs

Table 3. Top 20 most requested posts retrieved from the IAlogs

No.	Post URI-R	Account Username
1	https://instagram.com/p/-CzOWjmEUZ/	@madonna
2	https://instagram.com/p/-DSiwbAQdA/	@ichiroyamaguchi
3	https://instagram.com/p/6pzghQttTe/	@rockwallwomensleague
4	https://instagram.com/p/-C-NNrHZXh/	@joannsfar
5	https://instagram.com/p/-DHq5xlZtw/	@3jsb_hiroomi_tosaka
6	https://instagram.com/p/-FL7ijGWDf/	@jayspizzy
7	https://instagram.com/p/--_dMXx4Lw/	@blakelively
8	https://instagram.com/p/-IMlMPBEVB/	@rondarousey
9	https://instagram.com/p/7I5TTpo4M0/	@avrillavigne
10	https://instagram.com/p/-2HG73JjZT/	@chrissyteigen
11	https://instagram.com/p/-dXZzJpjRW/	@chrissyteigen
12	https://instagram.com/p/-EdwWcr_y3/	@charlotte49ers
13	https://instagram.com/p/-8vsO9Fz6f/	@instavideo_kaz
14	https://instagram.com/p/-d6F1LJeps/	@charliemay
15	https://instagram.com/p/-cPSKcS_d-/	@bydvnlln
16	https://instagram.com/p/-2g67gy_av/	@bydvnlln
17	https://instagram.com/p/-hwc34y_a2/	@bydvnlln
18	https://instagram.com/p/-EkMzcy_Te/	@bydvnlln
19	https://instagram.com/p/-9KlOQDpvP/	@designseeds
20	https://instagram.com/p/-7GsJljpph/	@designseeds

```
$ curl -Is https://www.instagr.am/
HTTP/2 301
location: https://www.instagram.com/
content-type: text/plain
content-length: 0
server: proxygen-bolt
x-fb-trip-id: 1679558926
date: Wed, 05 Apr 2023 11:02:29 GMT
```

Fig. 16. A curl request to `https://www.instagr.am/` redirects to `https://www.instagram.com/`

represents 0.1% of their dataset (280,292 URI-Rs out of 285,000,000 total URI-Rs).

From our IAholdings dataset, we filtered out 18,273 known URI-Rs for other Instagram services[1] including the ones listed in Table 4 that we identified. This resulted in 257,695 URI-Rs. We then filtered for URI-Rs that belong to Instagram posts (the ones that match the prefixes `https://instagr.am/p/` and `https://www.instagram.com/p/`) resulting in 192,359 post URI-Rs. We used a sample of these post URI-Rs to demonstrate extracting usernames from the live web and archived Instagram posts. First, we used these URI-Rs to extract the shortcodes for the Instagram posts, resulting in 2427 unique posts. When we obtained the data initially, all the URI-Rs in the sample were lowercase, which resulted in all the extracted shortcodes being in lowercase as well. Since Instagram shortcodes are case-sensitive, we needed to find a method to look for the shortcode with the proper case to construct our dataset. We found that we could use the response from CDX API with prefix match to obtain the shortcode with the correct case. In Figure 17, we have highlighted how the response from CDX API for the URI-R prefix "https://www.instagram.com/p/d/" with shortcode `d` will return the URI-Ms with the correct cased shortcode `D`. Appendix I includes the code used for this case correction step. We used this technique and made a query to the CDX API to obtain the final set of 2427 post URI-Rs.

To demonstrate how we can extract post data from live and archived Instagram account pages, we created two datasets. We built the KatyPerryLive dataset by scraping Instagram posts on Katy

[1] `https://github.com/himarshaj/MSThesis_ArchivedIGPosts/blob/main/Analysis/instagram.urls.csv`

```
$ curl "https://web.archive.org/cdx/search/cdx?url=https://www.instagram.
   com/p/d/&matchType=prefix"
com,instagram)/p/d/embed 20141210001423 http://instagram.com/p/D/embed/
   text/html 301 QH732FYSV7UM34JYWVYMB7EZGR2CYM6B 446
com,instagram)/p/d/embed 20150531054222 https://instagram.com/p/D/embed/
   text/html 200 JMBGQ4MVKZ5UPSEKVNBH7OLKNMVGSAKF 28232
com,instagram)/p/d/embed 20151119184309 http://instagram.com/p/D/embed/
   text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 549
com,instagram)/p/d/embed 20151125220318 https://instagram.com/p/D/embed/
   text/html 200 ABFZYHAUW4AH6RLDPTKU74WI6OII737Q 3485
com,instagram)/p/d/embed 20221207103946 https://instagram.com/p/D/embed/
   text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 673
```

Fig. 17. CDX API response for the URI-R prefix "https://www.instagram. com/p/d/" of shortcode `d` returning the URI-Ms with the correct cased shortcode `D`

Perry's Instagram account (`@katyperry`) from her account page on the live web. As of September 24, 2020, Katy Perry had a total of 1642 posts on her Instagram account, and we were able to extract all of the posts by capturing the network traffic in HTTP Archive (HAR) format while scrolling down her account page until the first post. We built the BlakeLivelyArchive dataset by scraping Instagram posts on Blake Lively's Instagram account (`@blakelively`) from URI-Ms of her account page. As of February 23, 2023, `@blakelively` had 124 posts in her account, and we were able to extract 70% of her posts (87 out of 124) by using 3854 URI-Ms of her account page. A detailed explanation involving the creation of these two datasets can be found in Section 4.3.3.

4.2 APPROACH 1: REVISIT RECORDS

Revisit records are a specific type of WARC record that is used for recording revisions to previously archived web resources. According to the definition of revisit records, they are optional WARC records that can be used for material cross-referencing as well as reducing storage size [142]. By using revisit records[2] , we can create a cross-referencing between the original post URI-R of the format `https://www.instagram.com/p/{shortcode}/` and the URI-R modified

[2]During the thesis defense, Dr. Sawood Alam, a Web and Data Scientist at the Internet Archive suggested using synthetic 301 responses would be a more effective alternative to revisit records.

Table 4. Different types of URI-Rs for Instagram services

No.	URL Path
1.	/accounts/login
2.	/data/qe_params
3.	/stories/
4.	/static/
5.	/graphql/query?
6.	/tv/
7.	/explore/tags/
8.	/reel/

with the username of the post owner as `https://www.instagram.com/{username}/p/{shortcode}/`.[3]

To exemplify the functionality of this approach, we will use a post with shortcode `D`. This post is from the Instagram account of Kevin Systrom, the Co-founder and former CEO of Instagram, who has the username `@kevin`. Figure 18 is an example of a WARC response record for the post URI-R `https://www.instagram.com/p/D/`. Figure 19 is an example of how the new WARC revisit record should be specified, with the `WARC-Target-URI` set to `http://instagram.com/kevin/p/D/`. Additionally, Figure 18 and Figure 19 also illustrate the distinction between the `Content-Length` of the two records, with the revisit record being significantly smaller than the response record. By generating a new revisit record that integrates the Instagram account usernames into the Target-URI, users who request the URI-M of `http://instagram.com/kevin/p/D/` will receive the original content at URI-M for `http://instagram.com/p/D/`. By using WARC revisit records that contain the username, it becomes feasible to carry out CDX prefix searches that would enable us to obtain all the archived posts of a specific Instagram user.

In order to demonstrate our approach, we have implemented a small test archive with Instagram mementos. The process of using revisit records to automatically enable prefix search for discovering Instagram posts belonging to a particular account is illustrated in Figure 20.

For this study, we used the top 20 most requested posts from our IAlogs dataset and their respective Instagram user accounts as shown in Table 3. Each of these URI-Rs had at least 50

[3]It should be noted that including the username in an Instagram post URL like this is valid, although it is not a widely used convention.

```
WARC/1.0
WARC-Date: 2019-10-25T02:36:18Z
WARC-Type: response
WARC-Record-ID: <urn:uuid:1c88a09f-8bd4-4cb9-82a0-69cc92078204>
WARC-Target-URI: https://www.instagram.com/p/D/
WARC-Payload-Digest: sha1:4WOIHRPWUI2C4MFSCAO2BSUG6GHTF3NH
WARC-Block-Digest: sha1:WKWBI6XEGAGCI7A6472MK2O7PXRPGNJG
Content-Type: application/http; msgtype=response
Content-Length: 20725
...
```

Fig. 18. An example WARC response record with the `WARC-Target-URI` as `https://www.instagram.com/p/D/`

mementos available in the Internet Archive. We compiled a list of 1000 URI-Ms using the CDX API, 50 URI-Ms for each post (20 URI-Rs x 50 URI-Ms = 1000 URI-Ms). For each of those 1000 URI-Ms, we synthesized WARC response records. As the next step, we used the usernames of the accounts that those posts belonged to and synthesized WARC revisit records for each of them by setting the `WARC-Target-URI` header as the post URI-R with the username included. For synthesizing response and revisit WARC records, we used the basic WARCWriter class of the WARCIO library [143] for writing to a single WARC file.

Finally, we indexed the WARCs and replayed them through PyWb [144]. Figure 21 shows the search page of the PyWb interface for the IA_insta collection. We could search for an Instagram post URI (`https://instagram.com/p/-DSiwbAQdA/`) and it would return the search result page as shown in Figure 22. There are 50 URI-Ms for the requested post URI and by clicking on a specific Memento-DateTime, the user could see the replayed Instagram post page which belongs to the user account `@ichiroyamaguchi` (Figure 24). As you can see on Figure 23, you could access the same URI-Ms for the post using the post URI that includes the username (`https://instagram.com/ichiroyamaguchi/p/-DSiwbAQdA/`) as well.

A user account that has multiple posts among the top 20 most requested posts in the IA access logs is used to illustrate how the prefix search can be used to discover URI-Ms for all the posts of that account. As listed in Table 3, user account `@bydvnlln` has four posts in our dataset. As highlighted in Figure 25, the search results page of the PyWb interface for the Instagram URI

```
WARC/1.0
WARC-Refers-To-Target-URI: http://instagram.com/p/D/
WARC-Date: 2019-10-25T02:36:18Z
WARC-Type: revisit
WARC-Record-ID: <urn:uuid:eb1e24ed-fc2c-4a81-a69b-386158701d8f>
WARC-Target-URI: http://instagram.com/kevin/p/D/
WARC-Block-Digest: sha1:BFKJCQIWFZSJH4NKFVHN3ZPACTEGF4SZ
Content-Type: application/http; msgtype=response
Content-Length: 4715
...
```

Fig. 19. An example WARC revisit record with the `WARC-Target-URI` with the username included (`https://www.instagram.com/kevin/p/D/`)

prefix `https://instagram.com/bydvnlln/p/*` shows all the 200 post URI-Ms (50 URI-Ms each for the four URI-Rs) for the user `@bydvnlln`.

PyWb also supports responses to CDX queries in CDXJ format [145]. As shown in Figure 26, the CDXJ response shows several URI-Ms with MIME type as WARC revisit record. The CURL request followed by a few commands as shown in Figure 27 shows a summary of the number of URI-Ms returned on the query matching the prefix `https://instagram.com/bydvnlln/` by using the `matchType` as `prefix`. We can see that there are 50 URI-Ms each for the posts with shortcode `-2g67gy_av`, `-cPSKcS_d-`, `-EkMzcy_Te`, and `-hwc34y_a2` all of which belong to the user `@bydvnlln`. The usage of revisit records demonstrates that it is possible to enable automatic support for prefix search, enabling the discovery of all post mementos in a web archive pertaining to a specific Instagram user account.

4.3 APPROACH 2: SECONDARY INDEX

In this section, we will discuss a different method for handling top-level Instagram post queries by building a secondary index to map each Instagram user with its post shortcodes. The techniques that we are proposing to fill the index are summarized in Figure 28. An interested user can obtain all post shortcodes for a specific user from this index and use them to separately query the Internet Archive for mementos of each post. The final goal is to make sure that every Instagram post in IA is mapped to the user it belongs to and kept in the index. To achieve this, we will gradually create

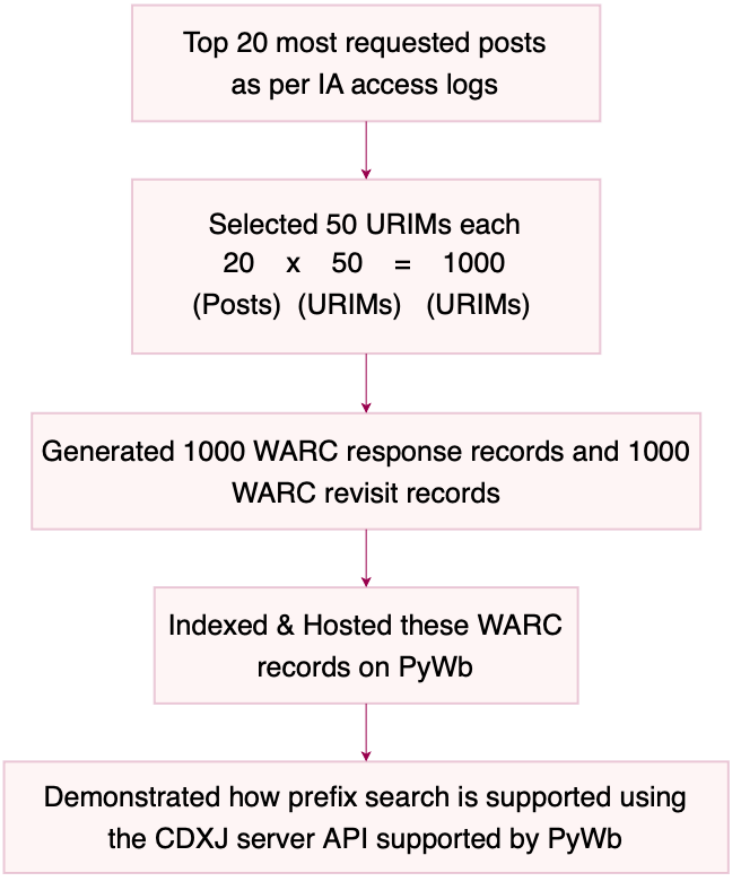

Fig. 20. Summary of the process involved in demonstrating prefix search for Instagram top-level post queries through the use of revisit records

this index using the following techniques:

1. A secondary JavaScript file, `unxtract.js`, that is used in addition to `wombat.js` upon replay (client-side)
2. Scraper scripts to extract the username from the Instagram posts

Before covering each of the various methods used to populate the index, it is crucial to comprehend how the index is built and what endpoints are supported.

4.3.1 Index

In order to store the collected `{shortcode,username}` pairs, we created a new database

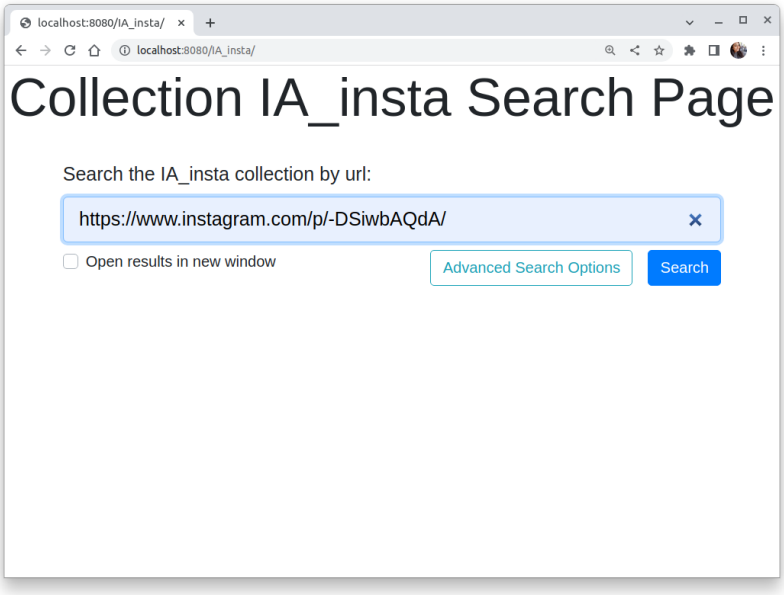

Fig. 21. Search page of the PyWb interface for the IA_insta collection

(namely `instagram.db`) using SQLite3[4] and a new database table (namely `IG_Index`). Figure 29 shows a screenshot of the database table having username, shortcode, and added_date as records, where the added_date is the `CURRENT TIMESTAMP` (reading of the time-of-day clock when the SQL statement is executed at the application server).

We developed a Flask[5] server using Python 3.0 with three API endpoints that can be used to read from and write to the IG_Index:

1. **add_un_idx**: This endpoint is used only by the client-side JavaScript method where the extracted username is directly sent to this endpoint to be added to the index. This will be discussed in detail under Section 4.3.2.

2. **get_posts**: This endpoint is used to obtain the post shortcodes that belong to a given Instagram user account by providing the username. The below CURL request shows how we can make a call to the API to get all posts related to the user account `@bydvnlln`.

```
$ curl -s http://127.0.0.1:5000/get_posts?username=bydvnlln
{"shortcodes":"-cPSKcS_d-,-2g67gy_av,-hwc34y_a2,-EkMzcy_Te",
"username":"bydvnlln"}
```

[4] `https://docs.python.org/3/library/sqlite3.html`
[5] `https://flask.palletsprojects.com/en/2.2.x/`

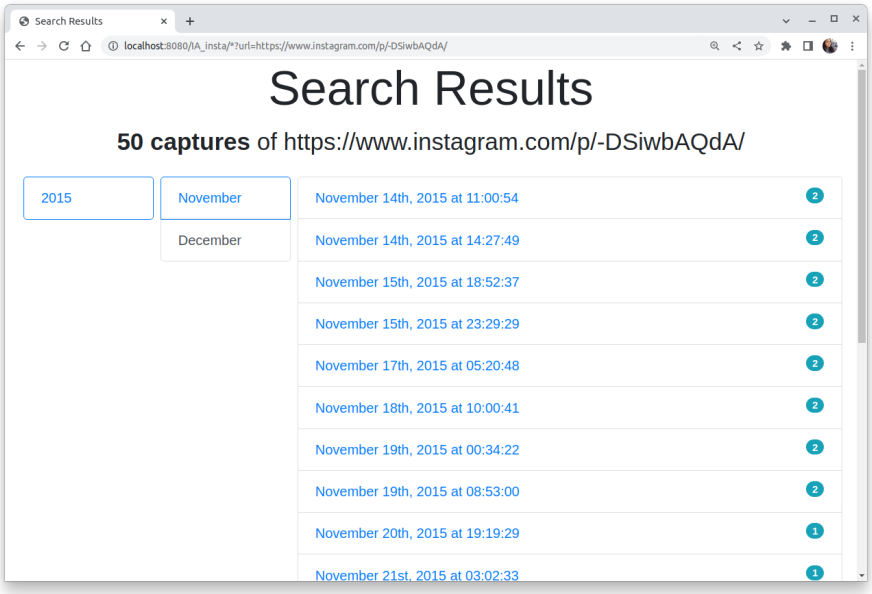

Fig. 22. Search results page of the PyWb interface for the Instagram post URI (`https://instagram.com/p/-DSiwbAQdA/`)

3. **get_username**: This endpoint is used to obtain the username of an Instagram user account by providing the shortcode of the post. The following CURL request shows how to use the API to retrieve the username of the Instagram account to which the post with shortcode `-CzOWjmEUZ` belongs.

```
$ curl -s http://127.0.0.1:5000/get_username?shortcode=-CzOWjmEUZ
{"shortcode":"-CzOWjmEUZ","username":"madonna"}
```

4.3.2 Client-Side JavaScript (unxtract.js)

In this approach, we propose introducing a second client-side JS library, in addition to wombat.js, at replay time. Whenever a user replays a memento of an Instagram post, this additional library will be responsible for recognizing it as an Instagram post and extracting the username from it. The shortcode of the post and the username should be added to the proposed index, IG_Index. Over time, we will be able to get many (not all) of the `{shortcode,username}` values (on average, the more popular ones).

To demonstrate the functionality of our proposed approach, we have implemented a basic and constrained version of the solution as a JavaScript library named `unxtract.js`. Appendix A

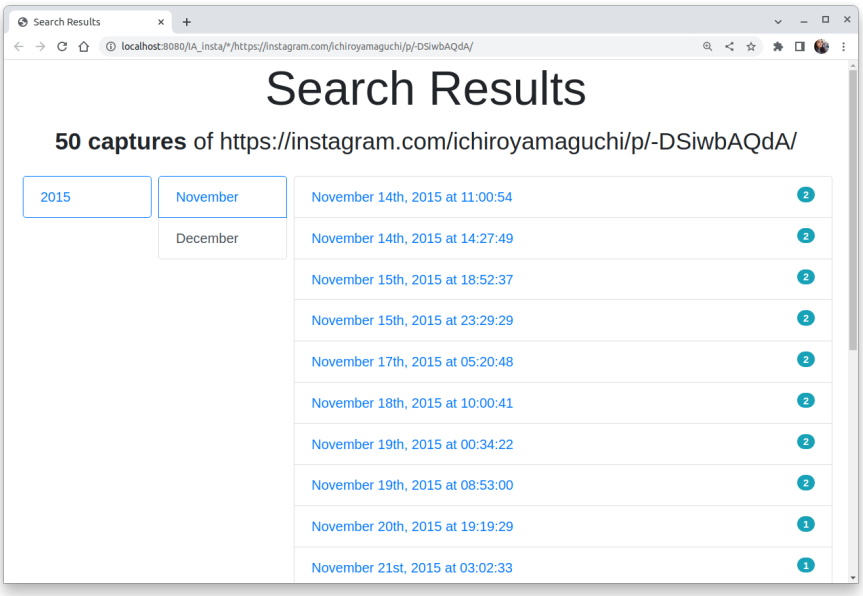

Fig. 23. Search results page of the PyWb interface for the Instagram post URI including the username (`https://instagram.com/ichiroyamaguchi/p/-DSiwbAQdA/`)

presents the `unxtract.js` code that we used to extract the `script` tag that contains the Instagram account username of the post owner. After extracting the value of the `script` tag, the JavaScript library will forward it to the `add_un_idx` endpoint of the API. At this endpoint, the username extraction process will take place before the user is added to the index. The code used for this process is available and explained in Appendix C. It should be noted that the data we are interested in, which is contained within the `script` tag, is formatted in JSON. Figure 30 provides an example JSON snippet for the post with the shortcode `--_dMXx4Lw`. As you can see, the username of the post owner (`@blakelively`) can be found inside the `owner` object. Appendix B provides a sample JSON snippet with more data.

To test this implementation, we used PyWb to build another test archive that is smaller in size but similar to the one developed in Section 4.2. We synthesized 100 WARC response records (five URI-Ms for each post) for the top 20 most requested posts from the IAlogs dataset listed in Table 3. The post with the shortcode `-8vs09Fz6f` will serve as our demonstration example. Figure 31 displays the PyWb search result page for the `https://www.instagram.com/p/-8vs09Fz6f/`. This post belongs to a user account `instavideo_kaz` but as can be seen below in Figure 32, it is not yet added to our index.

Consider a scenario where a user clicks on the URI-M from December 24th, 2015 in Figure 31.

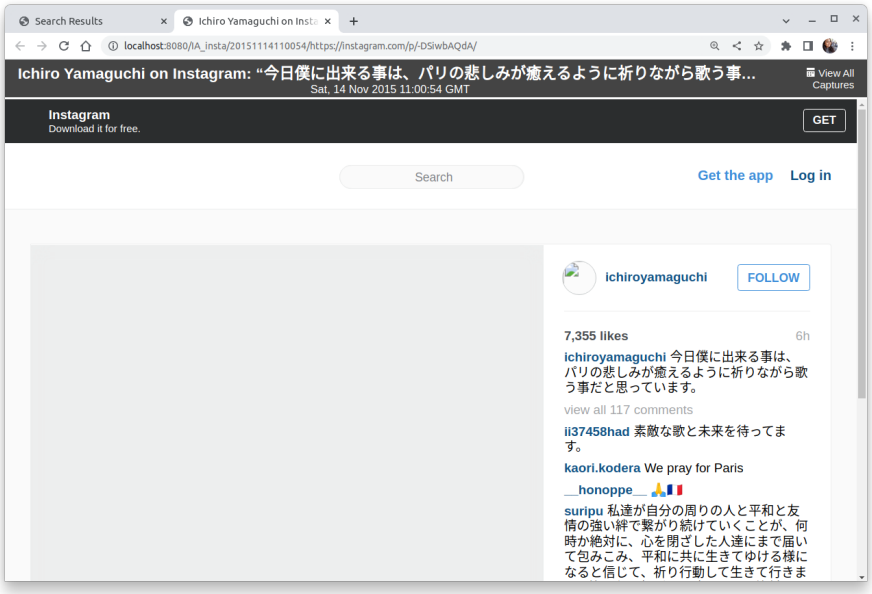

Fig. 24. A replayed Instagram post (`https://instagram.com/p/-DSiwbAQdA/`) on PyWb interface which belongs to the user `@ichiroyamaguchi`

During the replay of the memento, the system initiates the username extraction from the URI-M in the background and sends it to the previously introduced `add_un_idx` API endpoint. The proposed JavaScript (`unxtract.js`) will manage this task. As shown in Figure 33, `unxtract.js` is requested and it would extract the `script` tag that contains the username and initiate a POST request to the `add_un_idx` API endpoint Flask server as shown in Figure 34.

The addition of the username to the IG_Index is verified by looking into the content of the database table as shown in Figure 35. As shown in Figure 36, we can see that the API response indicates the username of the post owner as `@instavideo_kaz`

If the same URI-M or a different URI-M for the same post is requested, the server will recognize that it already exists in the index and respond stating that the `{username,shortcode}` pair already exists. Figure 37 shows the response when the URI-M from December 10th, 2015 is requested.

The above demonstration serves as evidence that we can employ client-side JavaScript during replay to continually populate our IG_Index as users browse through Instagram posts in web archives.

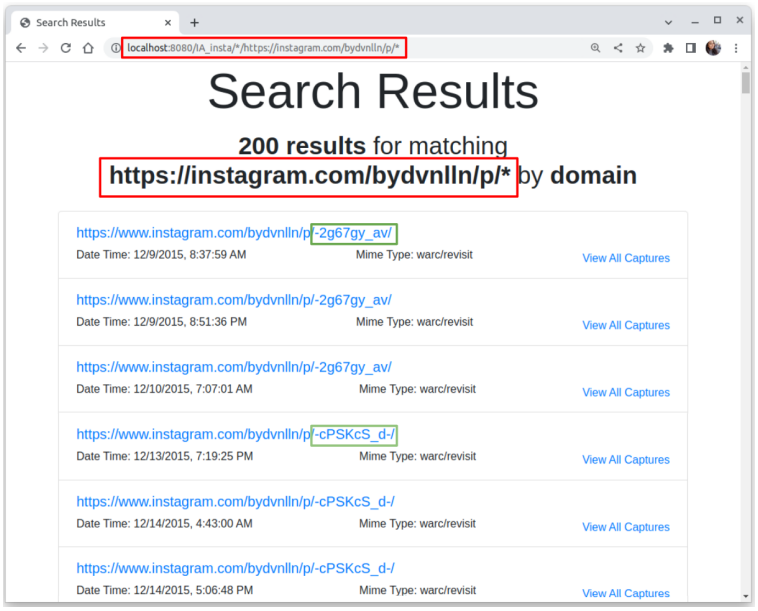

Fig. 25. Search results page of the PyWb interface for the Instagram URI prefix `https://instagram.com/bydvnlln/p/*` showing all the post URI-Ms for the user `@bydvnlln`

4.3.3 Extracting Instagram Usernames

In this section, we will explore alternative methods we can use for populating the IG_Index with the objective of collecting as many `{username,shortcode}` pairs for incorporation into the IG_Index. Each of the techniques entails using distinct scraper scripts to either extract usernames from post pages or extract shortcodes from account pages. We will examine each of these scraper scripts, classified as follows:

1. To extract usernames from post pages in the live web (URI-Rs).
2. To extract usernames from archived post pages (URI-Ms).
3. To extract shortcodes from account pages in the live web (URI-Rs).
4. To extract shortcodes from archived account pages (URI-Ms).
5. To extract usernames from WARC response record for Instagram post page.

```
$ curl -s "http://localhost:8080/IA_insta/cdx?url=https://www.instagram.
   com/bydvnlln/&matchType=prefix" | head -3
com,instagram)/bydvnlln/p/-2g67gy_av 20151209133759 {"url": "https://www.
   instagram.com/bydvnlln/p/-2g67gy_av/", "mime": "warc/revisit", "length
   ": "1475", "offset": "0", "filename": "rev-0
   c2832f22e240f15b9a209d57c67cd3d-20211002121027-bydvnlln.warc.gz", "
   source": "IA_insta/indexes/index.cdxj", "source-coll": "IA_insta", "
   access": "allow"}
com,instagram)/bydvnlln/p/-2g67gy_av 20151210015136 {"url": "https://www.
   instagram.com/bydvnlln/p/-2g67gy_av/", "mime": "warc/revisit", "length
   ": "1519", "offset": "0", "filename": "rev-9711237
   a40dd327005c9adcc62af06f8-20211002121028-bydvnlln.warc.gz", "source": "
   IA_insta/indexes/index.cdxj", "source-coll": "IA_insta", "access": "
   allow"}
com,instagram)/bydvnlln/p/-2g67gy_av 20151210120701 {"url": "https://www.
   instagram.com/bydvnlln/p/-2g67gy_av/", "mime": "warc/revisit", "length
   ": "1517", "offset": "0", "filename": "rev-78
   abe1b4416f7fd030854c755c87abd4-20211002121031-bydvnlln.warc.gz", "
   source": "IA_insta/indexes/index.cdxj", "source-coll": "IA_insta", "
   access": "allow"}
```

Fig. 26. The CDXJ response to the CURL request showing several URI-Ms with MIME type as WARC revisit record that matches the prefix "`https://instagram.com/bydvnlln/`"

We used prominent libraries such as Beautiful Soup,[6] requests,[7] re,[8] and JSON[9] for creating these scraper scripts in Python programming language.

[6] `https://beautiful-soup-4.readthedocs.io/`
[7] `https://requests.readthedocs.io/`
[8] `https://docs.python.org/3/library/re.html`
[9] `https://docs.python.org/3/library/json.html`

```
$ curl -s "http://localhost:8080/IA_insta/cdx?url=https://www.instagram.
   com/bydvnlln&matchType=prefix" | awk -F'"' '{print $4}' | sort | uniq -
   c
    50 https://www.instagram.com/bydvnlln/p/-2g67gy_av/
    50 https://www.instagram.com/bydvnlln/p/-cPSKcS_d-/
    50 https://www.instagram.com/bydvnlln/p/-EkMzcy_Te/
    50 https://www.instagram.com/bydvnlln/p/-hwc34y_a2/
...
```

Fig. 27. Summary of the number of URI-Ms returned on the query matching the prefix "https://instagram.com/bydvnlln/"

Extract Usernames from Post Pages in the Live Web (URI-Rs)

In this method, we extracted the usernames from post pages on the live web. We used the IAholdings dataset, the Instagram holdings sample from the Internet Archive introduced in Section 4.1, for demonstration purposes.

For all of the 2427 unique shortcodes extracted from Instagram post URLs in the dataset, we scraped the live web and downloaded the HTML file for a not-logged-in user. We noticed that the not-logged-in version of the HTML page contains more metadata than the logged-in version of the post page HTML. Additionally, we found that the not-logged-in version is more accessible and provides a consistent view of the page that is available to all users. As Instagram will temporarily redirect the requests to its login page after a couple of requests, this scraper script is designed to wait until the penalty period is completed, and then continue collecting the data. After having exhausted different methods to avoid getting detected by Instagram, we took the help of our research group members to collect this data in a distributed manner using different machines. We divided the 2427 post URLs into files of 25 post URLs each (98 parts) and used those as input files to the scraper script. This approach enabled us to overcome several challenges (detailed in Section 4.4) and expedite the data collection process.

The scraper script takes a list of shortcodes for Instagram posts as input and generates several output files. For example:

- **shortcodeaa**: A text file containing the shortcodes of Instagram posts to be downloaded. This is one of the 25 input parts

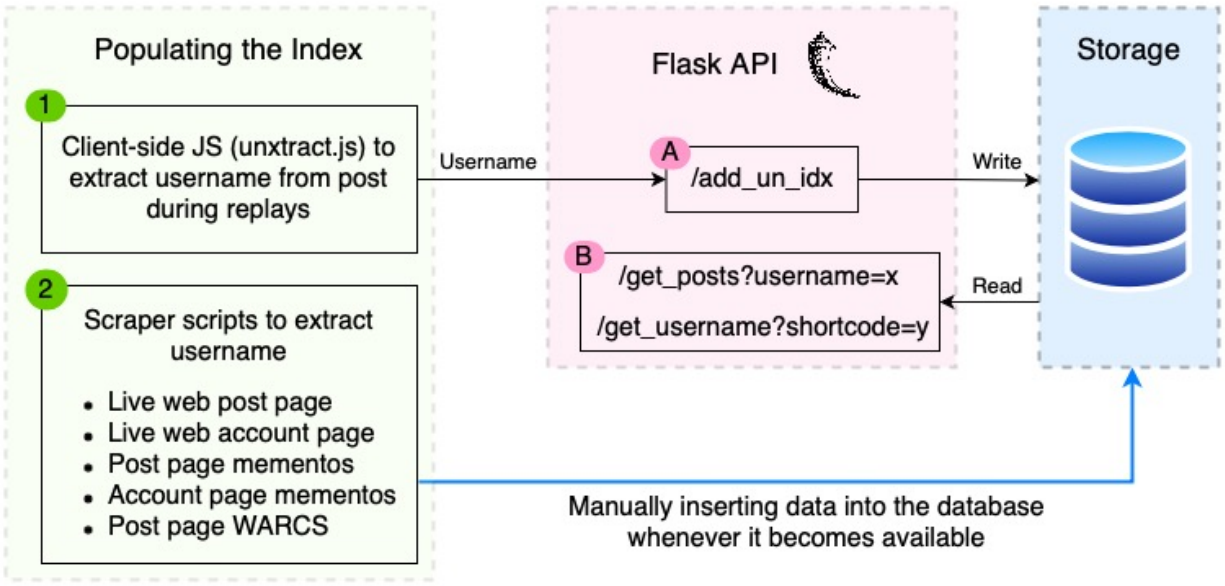

Fig. 28. The techniques that can be used to populate the proposed secondary index. The index is stored in a database and accessed via a Flask API for seamless read and write operations

- **{shortcode}.html** - The downloaded HTML files for each post in the input file
- **shortcodeaa_out.tsv** - A summary file containing the response status code and headers for each post in the input file
- **shortcodeaa_blockdata.csv** - A CSV file containing information about the penalty period
- **shortcodeaa_requests.log** - A log file containing request logs

Once the HTML files have been downloaded for each post, we extracted the username from each of them using a different scraper script. We used a trial-and-error method to look for the username at different locations in the HTML. Figure 38 illustrates `{name="twitter:title"}` and `{property="og:description"}`, the two `meta` fields that can be used to extract the username. The code used to extract the username is shown in Appendix D.

Extract Username from Archived Post Pages (URI-Ms)

Out of the 2427 posts in the IAholdings dataset, 34.36% (833) posts are no longer available on the live web. For those that are not available from the live web, we used the archive to obtain the usernames from post URI-Ms. In this approach, we wrote a scraper script to parse mementos of

```
Successfully connected to the instagram.db Database!

Database table: IG_Index

              username   shortcode           added_date
0              madonna  -CzOWjmEUZ  2022-11-11 15:52:04
1      ichiroyamaguchi  -DSiwbAQdA  2022-11-11 16:22:23
2  3jsb_hiroomi_tosaka  -DHq5xlZtw  2022-11-11 16:23:28
3          blakelively  --_dMXx4Lw  2022-11-11 16:24:01
4         avrillavigne  7I5TTpo4M0  2022-11-11 16:24:16
```

Fig. 29. Screenshot displaying the content added to the IG_Index database table with a column for username, shortcode, and added_date

Instagram posts to extract the username of the post owner. Like many other websites, Instagram's UI has undergone numerous design modifications over time, making it challenging to extract the username from Instagram posts by using a single rule. We overcame this difficulty by using various cases to manage these different UI designs based on the Memento-Datetime, which enabled us to scrape the posts to extract the usernames. In Table 5, we have listed the different cases we used to retrieve the username based on the different time ranges. The rules and categories have been identified through manual observation and trial and error and are based on our current understanding.

We used the 833 posts (out of 2427 posts in the IAholdings dataset) that are no longer available on the live web to demonstrate how the usernames can be extracted from the post page URI-Ms. For the 833 posts, we downloaded the CDX response with all the mementos and saved it in a file (one text file for each post). There were a total of 1,212,228 mementos for the 833 posts. The scraper script iterated through each of the post URLs and looked for the text file containing the list of URI-Ms for that particular post. It read the list of mementos, grabbed the `Memento-Datetime` and generated a list of mementos ordered according to the date. The scraper script then tried to extract the username from the earliest memento and kept trying until it was able to extract the username and finally write the output to a file with the shortcode, extracted username, and the URI-M used. We made the decision to start from the earliest memento, as we have uncovered that almost all of the recent Instagram mementos are redirecting to the account login page [14, 15] and there is no username to extract. This choice guarantees that we can extract the username from mementos that are more likely to contain it early on, rather than having to browse through the mementos randomly.

While doing this process, we also found that more than 78% (946,248 out of 1,212,228) of

```
{"entry_data": {
   "PostPage": [{
      "__query_string": "?",
      "media": {
        "code": "--_dMXx4Lw",
        "owner": {
          "username": "blakelively",
          "full_name": "Blake Lively",
          "requested_by_viewer": false,
          "followed_by_viewer": false,
          "has_blocked_viewer": false,
          "profile_pic_url": [...],
          "is_unpublished": false,
          "blocked_by_viewer": false,
          "id": "1437529575",
          "is_private": false},
        "comments": {...},
        "likes": {...},
        "date": 1449477651,
        "is_video": false,}]},
 "hostname": "www.instagram.com",
 "config": {...},
 "environment_switcher_visible_server_guess": true }
```


Fig. 30. An example JSON snippet for the post with the shortcode `--_dMXx4Lw`. The username of the post owner (`@blakelively`) can be found inside the `owner` object

the URI-Ms are embedded (Figure 39) or media (Figure 40) URLs. Because the mementos of the embedded URLs still contain the username, we incorporated this case in the scraper script. However, we found that most of the media URLs redirect to a CDN service and contain only the post image and no username information to be extracted. Out of the 1,212,228 mementos for the 833 posts, 685,552 (56.55%) were media URI-Ms. With the help of a Python program, we randomly selected 100,000 media URI-Ms and sent requests to each one of them. Out of

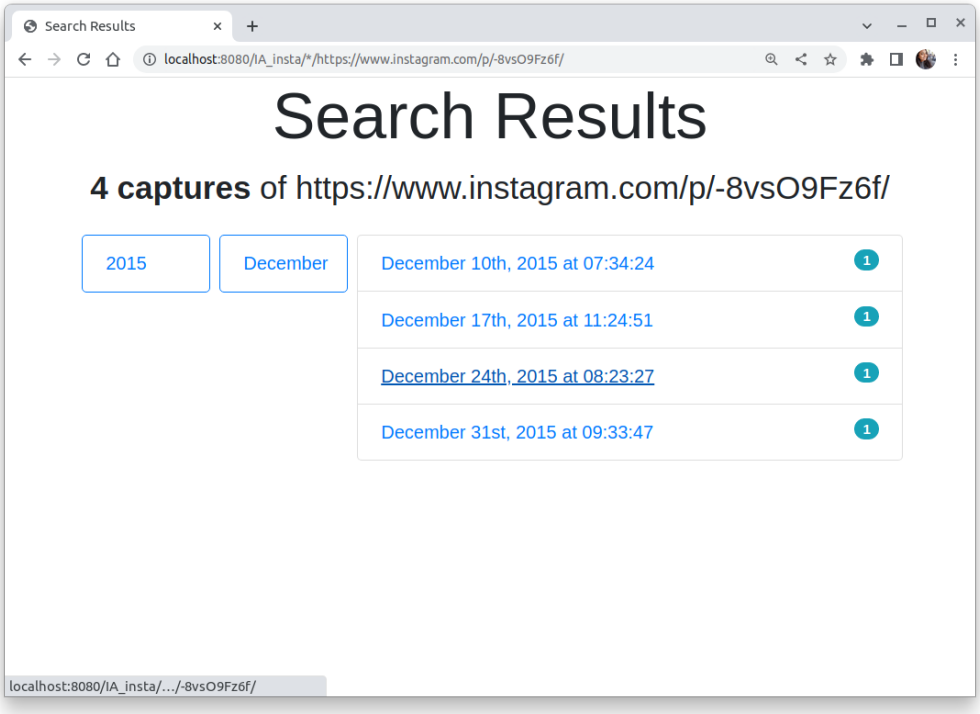

Fig. 31. The PyWb search result page for a post with the shortcode `-8vsO9Fz6f`. There are four captures for the URI-R `https://www.instagram.com/p/-8vsO9Fz6f/`

the 100,000 URI-Ms, we discovered that 96,283 (96%) are redirects. The remaining URI-Ms are mostly captures of the Instagram error page (archived 404s). Out of the 96,283 URI-Ms, 95,447 (99%) redirect to known CDN services (Figure 41). The remaining 1% are either redirects to the Instagram login page or a few of the other CDN services, which are much less prevalent and are, therefore, harder for our code to identify.

Extract Shortcodes from Account Pages in the Live Web (URI-Rs)

To demonstrate the process of obtaining shortcodes for a particular account from the live web, we will use the Instagram account of Katy Perry (`@katyperry`). As of September 24, 2020, Katy Perry had a total of 1642 posts on her Instagram account.

After exhausting several different approaches to collect the individual post data from the live web, we were able to use the Chrome developer tool to capture the network traffic in HTTP Archive (HAR) format while scrolling down until the first post in her Instagram feed. This approach is based on a method used in one of the preliminary studies to check how well is Instagram

```
$ curl -iLs http://127.0.0.1:5000/get_username?shortcode=-8vsO9Fz6f
HTTP/1.0 404 NOT FOUND
Content-Type: application/json
Content-Length: 44
Access-Control-Allow-Origin: *
Server: Werkzeug/1.0.1 Python/3.8.10
Date: Tue, 10 Jan 2023 18:15:29 GMT

{"shortcode":"-8vsO9Fz6f","username":null}
```

Fig. 32. A CURL request and response demonstrating the absence of the username for the post with the shortcode `-8vsO9Fz6f` in the IG_Index

archived [146]. A snippet of HAR data highlighting the field for extracting the shortcode is illustrated in Figure 42. More details on the HAR file are available in Appendix E. We used Haralyzer[10] to extract the relevant data fields that are required to compile the dataset[11] of her individual posts. By using this method, we were able to collect all the shortcodes for the 1642 posts on her account page at the time.

Extract Shortcodes from Archived Account Pages (URI-Ms)

In order to demonstrate how we can use mementos to extract shortcodes for a particular user, we will use the account of Blake Lively (`@blakelively`). As of February 23, 2023, `@blakelively` had 124 posts on her account. As the first step, we collected all the URI-Ms for the account page URI-R (`https://www.instagram.com/blakelively/`) using the prefix search of IA's CDX API (we performed this query on February 23, 2023) as shown below:

```
$ curl -s "http://web.archive.org/cdx/search/cdx?url=https://www.
   instagram.com/blakelively/&matchType=prefix" | awk '{print "https
   ://web.archive.org/web/" $2 "/" $3};'
```

There were 3854 URI-Ms and we used them as the input to a scraper script that extracted shortcodes for each of the posts in each URI-M. We used the requests library to make a request to

[10] https://haralyzer.readthedocs.io/

[11] `https://github.com/himarshaj/MSThesis_ArchivedIGPosts/tree/main/KatyPerryLive`

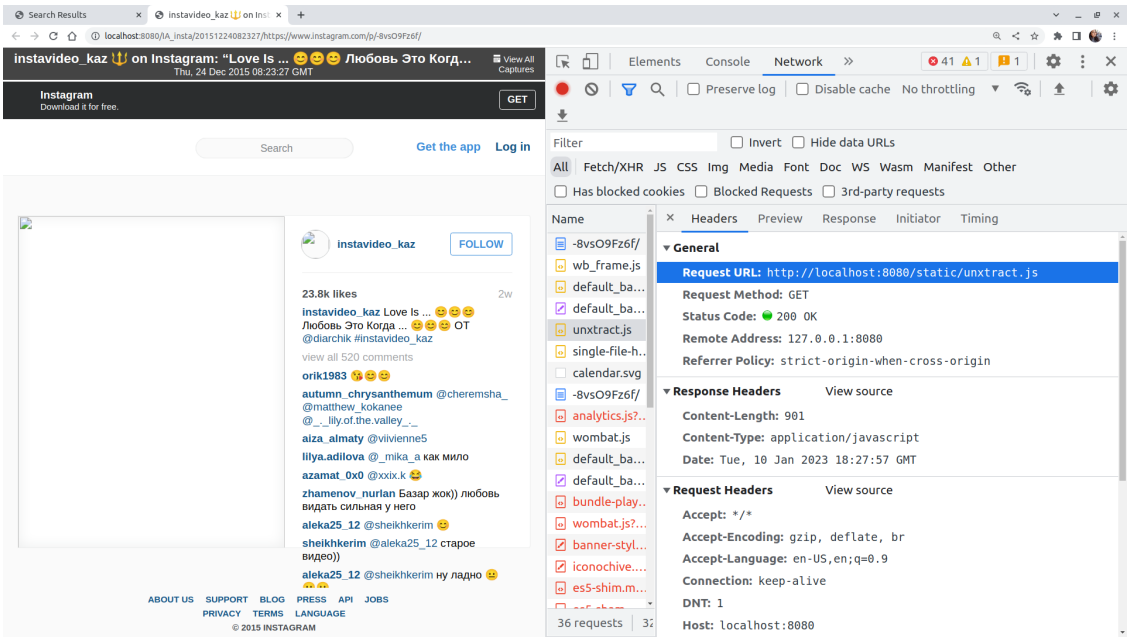

Fig. 33. A screenshot highlighting a GET request made to the newly introduced JavaScript (`unxtract.js`)

the URI-M and obtained the response. The post data is available in the response in JSON format inside a script tag as shown in Figure 43. We extracted the content in the script tag and used the JSON library to obtain the shortcode for each post available. Appendix F includes a JSON snippet with more details about where the post data is available. The code we used to extract the post data from account URI-Ms is shown in Appendix G.

Out of the 3854 mementos, we were able to extract posts from 2655 URI-Ms. By extracting the shortcodes for all the collected posts, we identified 87 unique shortcodes.[12] This means that we were able to collect 70% (87 out of 124) of the posts that belonged to the `@blakelively` account by using mementos.

We followed the same method for `@naomirwolf` but were only able to collect 26% (22 out of 84) of her posts. This is because `@naomirwolf` only had 106 URI-Ms for her account page as of February 23, 2023. Furthermore, we could only extract 12 posts from each account page memento. If the account is frequently archived and the frequency of posting by the user on their Instagram account is low, it is likely that we would be able to collect more posts that belong to that particular user. We note that Instagram employs lazy loading and requests additional images if the

[12] `https://github.com/himarshaj/MSThesis_ArchivedIGPosts/tree/main/BlakeLivelyArchive`

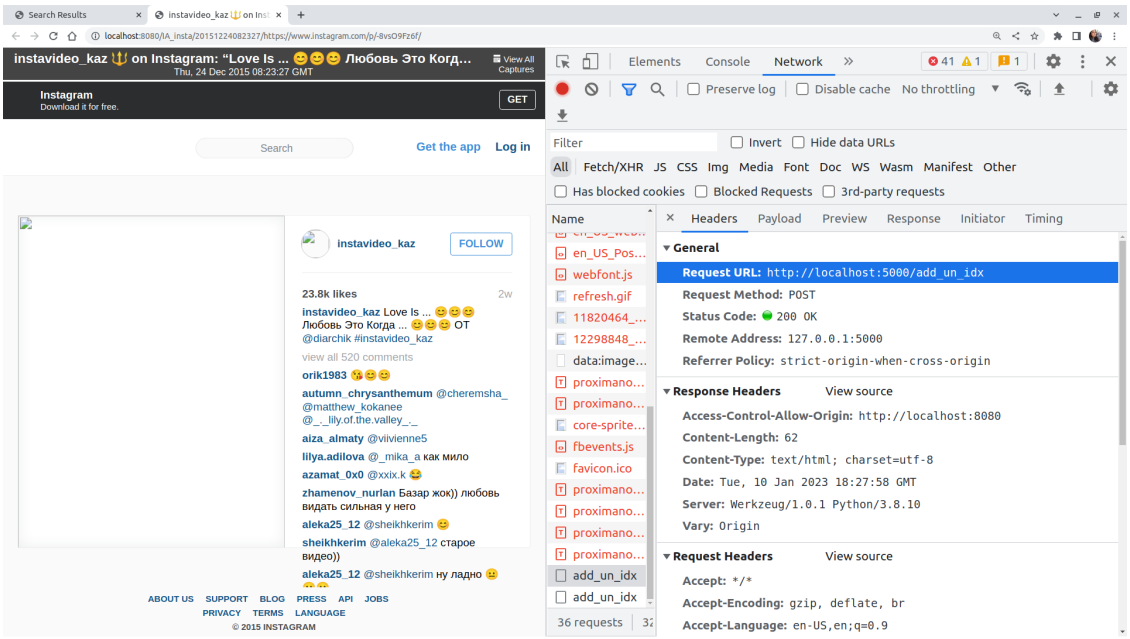

Fig. 34. A screenshot highlighting a POST request made to the `add_un_idx` API endpoint

page is scrolled. A headless browser on the script would have allowed us to automate the process and scroll down the page; however, we have observed that Instagram prompts us to log into the account if we scroll further so we have not implemented this method.

Extract Username and Shortcode from WARC Records

In this section, we demonstrate how we are able to extract the post shortcode and the username of the account that it belongs to from the WARC response record of a post URI-M. We are able to use the `WARC-Target-URI` field that contains the URI-R of the post to extract the shortcode of the post. We were also able to extract the username of the post owner by using the JSON data available in one of the `script` tags in the WARC response record payload.

An example WARC record for the post with shortcode `D` that belongs to the user `@kevin` is shown in Figure 44. We have highlighted the locations where the username and shortcode are available. Additionally, we noticed some cases where the username of the post owner is available in the `script` tag (Figure 44) and other cases which include the id of the user account in addition to the username (Figure 45).

We used the WARCIO library [143] to read a single WARC file and access the `WARC-Target-URI` (Figure 46). The value of the `WARC-Target-URI` is the post URI-R (`https://www.instagram.`

```
Successfully connected to the instagram.db Database!

Database table: IG_Index

              username   shortcode           added_date
0              madonna  -CzOWjmEUZ  2022-11-11 15:52:04
1      ichiroyamaguchi  -DSiwbAQdA  2022-11-11 16:22:23
2  3jsb_hiroomi_tosaka  -DHq5xlZtw  2022-11-11 16:23:28
3          blakelively  --_dMXx4Lw  2022-11-11 16:24:01
4        avrillavigne  7I5TTpo4M0  2022-11-11 16:24:16
5       instavideo_kaz  -8vsO9Fz6f  2023-01-10 18:27:58
```

Fig. 35. Screenshot displaying the username `instavideo_kaz` added to the IG_Index database table through client-side JS

`com/p/D/`) and the shortcode `D` is extracted from it.

We used regular expressions as shown in Figure 47 to match and extract the username from the payload in different scenarios. The first regular expression can be used to match the username in one step. We used the other two regular expressions to first extract the user id of the post owner and then use the user id in the second pattern to match and extract the username of the account owner.

The methods explained above allowed us to extract the shortcode of the post as well as the username of the post owner. More information on the code used for the extraction is available in Appendix H. We were able to use this script to collect the `{username,shortcode}` pairs from WARC files in our test sample to be added to the index.

We recognize that in a practical scenario, WARC files will not consist of a single record, but rather files containing multiple WARCs within them. We have not evaluated the tradeoffs involved in processing such large files at this stage.

4.4 DISCUSSION

Earlier in this chapter, we introduced our proposed approaches to support querying Instagram posts that belong to a specific account and provided proof of concept through implementation. In this section, we briefly discuss the feasibility of the proposed techniques and the resources they would demand. In addition, we will present a detailed discussion of the challenges we encountered

```
$ curl -iLs http://127.0.0.1:5000/get_username?shortcode=-8vsO9Fz6f
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 55
Access-Control-Allow-Origin: *
Server: Werkzeug/1.0.1 Python/3.8.10
Date: Tue, 10 Jan 2023 18:33:07 GMT

{"shortcode":"-8vsO9Fz6f","username":"instavideo_kaz"}
```

Fig. 36. A CURL request and response demonstrating the presence of the username for the post with the shortcode `-8vsO9Fz6f` in the IG_Index. We can see from the response that the post belongs to the account `@instavideo_kaz`

during data collection using our live web scraping methods and some interesting insights from the IAlogs dataset that we used for various implementations in our study.

4.4.1 Overview of Each Approach and Its Corresponding Requirements

We have proposed several new approaches to accessing Instagram posts related to a specific user in web archives. In this section, we will discuss the resources and requirements needed for the approach to function effectively. In this regard, there are several key questions to ask when evaluating a proposed approach including factors such as storage, computational power, and external dependencies. By considering these factors, web archivists and developers can better understand the feasibility and potential challenges of implementing these proposed approaches. The resources needed by various approaches are summarized in Table 6, based on the requirements:

1. **Archive Support:** This refers to whether the proposed approach is dependent on the support of the archive for it to work effectively. In other words, does the archive need to provide certain functionalities or services to make the proposed approach viable?

2. **Archive Storage:** This refers to whether the proposed approach requires storage at the archive. Will the archive need to store any additional data to facilitate the functioning of the approach?

3. **Archive Computational Power:** This refers to whether the proposed approach requires

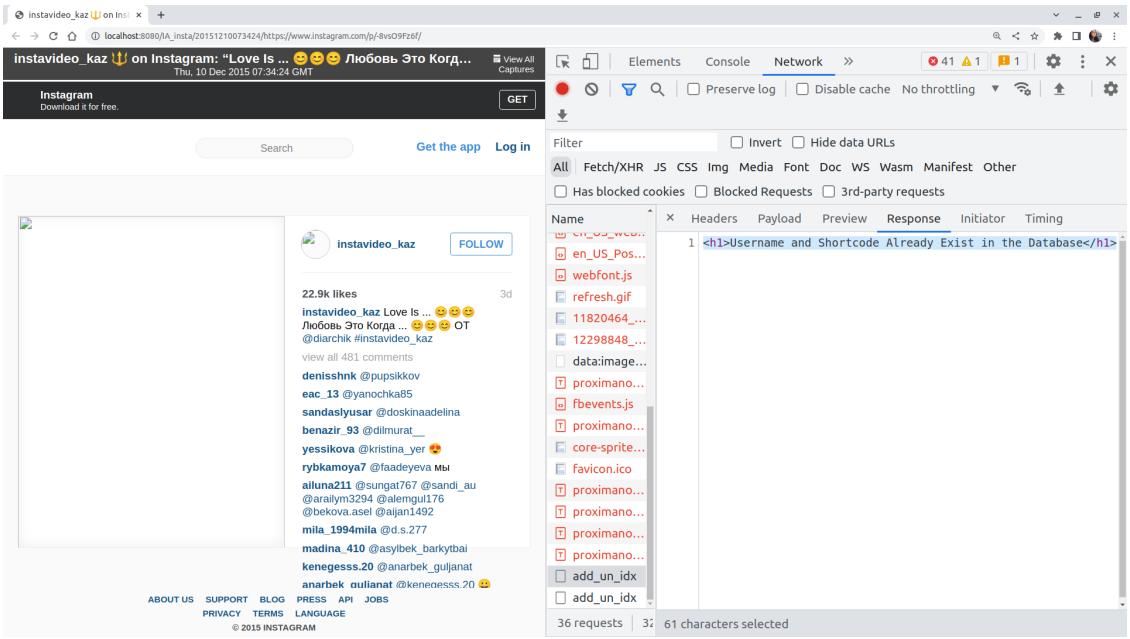

Fig. 37. Response from the `add_un_idx` API endpoint with the successful message

computational power from the archive. Will the archive need to perform any processing during archiving or replay to make the approach work?

4. **External Storage:** This refers to whether the proposed approach necessitates external storage outside of the archive. Will the approach require additional storage resources that are not provided by the archive?

5. **Live web Instagram:** This refers to whether the proposed approach requires any interaction or involvement with Instagram. Does the approach require any data or interaction with live web Instagram?

6. **Extra step:** This refers to whether an additional step is required for web archive users to make use of the proposed approach. Will users of the web archive need to perform any extra actions or follow any specific procedures to use the approach effectively?

The columns in Table 6 represent different factors or requirements, including whether the approach requires archive support, archive storage, archive computational power, external storage, an additional step for the user, or any data or interaction with live web Instagram. Table 6 includes several rows, each representing a unique proposed approach. Notably, two primary approaches are outlined in the table: the revisit record approach and the secondary index approach. The revisit

```
<!DOCTYPE html>
<html lang="en" class="no-js not-logged-in client-root">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title> [...] </title>
    <meta name="robots" content="noimageindex, noarchive">[...]
    <meta content=[...] />
    <meta property="og:site_name" content="Instagram"/>
    <meta property="og:title" content=[...]/>
    <meta property="og:image" content=[...]/>
    <meta property="og:description" content="57.5k Likes, 1,280 Comments
       - Lena Dunham (@lenadunham) on Instagram: "Ok I'm not in the habit
        of breaking promises to you people so here I am in the thrilling
       negligee" "/>
    <meta property="fb:app_id" content="124024574287414"/>
    <meta property="og:url" content="https://www.instagram.com/p/0
       G1rDoC1G6/"/>
    <meta property="instapp:owner_user_id" content="13875723"/>
    <meta name="twitter:card" content="summary"/>
    <meta name="twitter:site" content="@instagram"/>
    <meta name="twitter:image" content=[...]/>
    <meta name="twitter:title" content="Lena Dunham (@lenadunham) .
       Instagram photo"/>
    <meta name="twitter:maxage" content="86400"/> [...]
  </head>
  <body class="" style="background: white;">[...]</body>
</html>
```

Fig. 38. An HTML code snippet illustrating the {name="twitter:title"} and {property="og:description"}, the two meta fields that can be used to extract the username.

Table 5. The different cases we used to retrieve the username based on the different time ranges for archived post page URI-Ms

No.	Date Range	Location

1	2012-01-02 to 2014-01-25	`meta, property="og:description"`
		if embed:
		`script → "window._sharedData"`

2	2014-01-25 to 2015-01-10	`h2 → 'rel':'author'`
		if embed:
		`script → window._sharedData`

3	2015-01-10 to 2017-01-01	`meta, property="og:description"`
		if embed:
		`'class':'ehInfoUsername'`

4	2017-01-01 to 2020-12-31	`meta, property="og:description"`
		if embed:
		`'span',"class":"UsernameText"`

record-based approach necessitates the cooperation and assistance of the web archive in generating revisit records for each archived Instagram post. As a result, additional storage space is required to house these revisit records, although they are relatively smaller in size compared to the WARC response records. For all secondary index-based approaches, external storage is needed to accommodate the index. Additionally, users must first make an API call to obtain all the post shortcodes belonging to a given user from the index (to construct the URI-Rs) before querying the archive for the URI-Ms. The client-side JavaScript requires support from the web archive to incorporate our JavaScript file into their replay system. Although no considerable storage is required, the computational power for rendering pages to execute the JavaScript and send a POST request to our API is required. We suggested various methods for populating the index, which involved extracting usernames from posts and collecting all posts from account pages. Obtaining the URI-Ms necessary for extracting data from archived posts and accounts would require making an excessive number of calls to the archive. Additionally, when performing live web scraping, we would need to navigate around the obstacles posed by Instagram's strict rate-limiting measures, some of which are discussed in Section 4.4.2. To extract usernames from WARC records, we would require support from web archives as well as their computational power. It is both possible and practical to extract

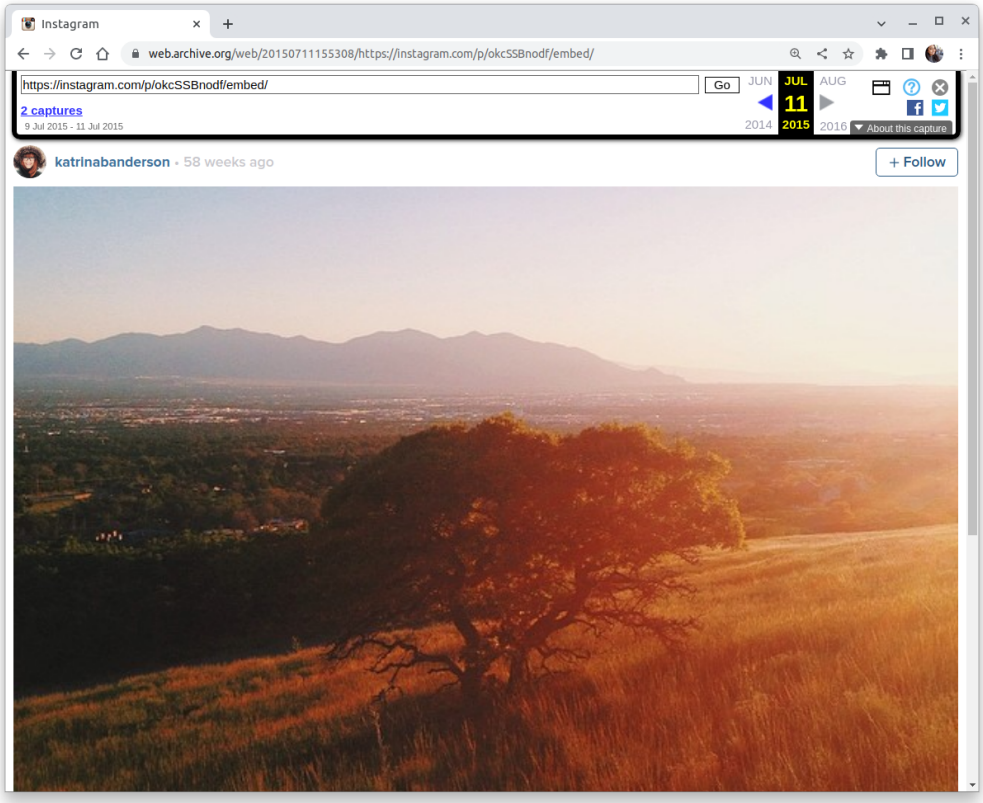

Fig. 39. An example embed URI-M - `https://web.archive.org/web/20150711155308/https://instagram.com/p/okcSSBnodf/embed/`

usernames directly from WARC records by requesting a limited number of large web archives to run our scraper script and provide us with the output, which can then be incorporated into the index.

4.4.2 Challenges Faced When Collecting Data From the Live Web

Obtaining content from Instagram was a challenging task due to its stringent rate-limiting policy. Additionally, the penalty times (block periods) after being blocked did not follow a discernible pattern. In order to demonstrate the technique of extracting usernames from live web post pages, which is discussed in Section 4.3.3, we used the unique post URIs in the IAlogs dataset. Despite attempting various methods to collect the HTML for those posts, we were unsuccessful and encountered several obstacles, which are outlined below.

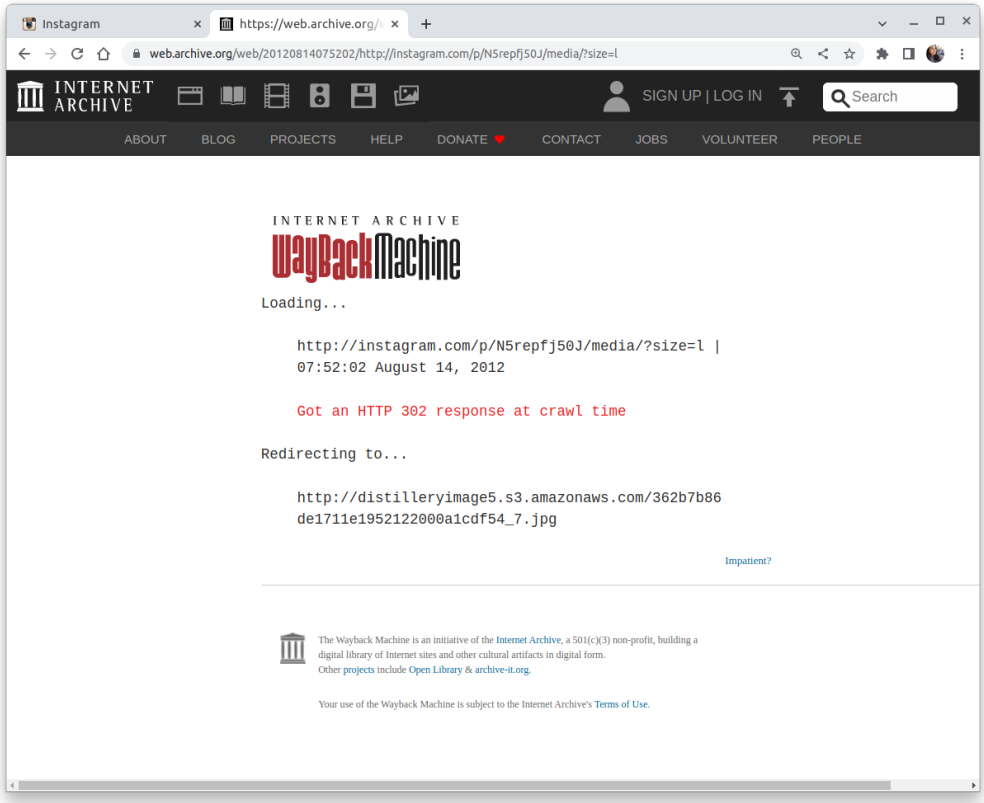

Fig. 40. An example media URI-M - `https://web.archive.org/web/20120814075202/http://instagram.com/p/N5repfj50J/media/?size=l`

- As stated in Section 2, the two new Instagram APIs are geared towards marketing research and not academic research and, thus, did not provide the data we needed.
- Despite attempting various methods to bypass Instagram's strict and unpredictable rate-limiting, we were unsuccessful. None of these methods (applied independently and in various combinations) proved effective, and the scraper was consistently redirected to the login page after only a few requests.
  - The addition of sleep times, which is a commonly used technique to bypass rate-limiting, did not prove to be effective in this case. Even after waiting for extended periods, the scraper was still redirected to the login page after only a few requests.
  - We tried the technique of rotating the User-Agent (UA) which involves changing the User-Agent HTTP request header to make it appear as if different users are making the

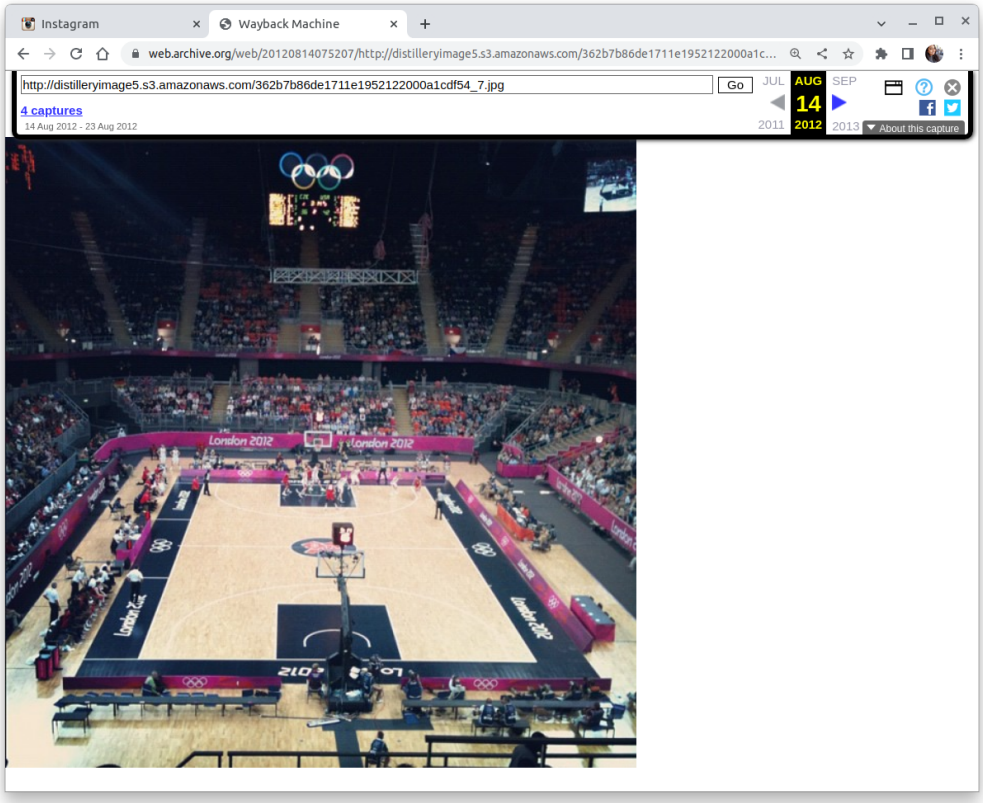

Fig. 41. The ultimate URI-M resulting from the redirection of the media URI-M (`https://web.archive.org/web/20120814075207/http://distilleryimage5.s3.amazonaws.com/362b7b86de1711e1952122000a1cdf54_7.jpg`). This is a URI-M of a CDN URL

requests each time.

- In addition, we attempted to use free proxy IP services as well as the Tor [147] browser through the command-line interface (CLI), but unfortunately, neither of these methods allowed us to bypass the rate-limiting.

However, we were able to obtain live web data by executing our code from various IP addresses in a distributed manner. As discussed in Section 4.3.3, each input file contained 25 posts and it took approximately 18-24 hours to complete one input file, but the time varied significantly due to factors such as Internet service provider, network connectivity, and type of Internet connection. Mobile data was faster than home Internet because most of the time when using mobile data we could disconnect and reconnect to obtain a new IP address. With the help of our research group,

```
{ "shortcode": "CEav2jYB12H",
   "edge_media_to_comment": {...},
   "edge_media_to_sponsor_user": {...},
   "comments_disabled": false,
   "taken_at_timestamp": 1598585366,
   "edge_media_preview_like": {...},
   "owner": {
      "id": "407964088",
      "username": "katyperry" },
   "location": {...} }
```

Fig. 42. A snippet of HAR data highlighting the field for extracting the shortcode `CEav2jYB12H`

Table 6. An overview of each approach and its corresponding requirements.

Approach	Archive Support	Archive Storage	Archive Computational Power	External Storage	Additional Step for User	Instagram
1) Revisit Records	yes	yes	yes	-	-	-
2) Secondary Index						
Client Side JS	yes	-	yes	yes	yes	-
Scraper Scripts						
(Live) Posts	-	-	-	yes	yes	yes
(Live) Accounts	-	-	-	yes	yes	yes
(Archive) Posts	-	-	yes	yes	yes	-
(Archive) Accounts	-	-	yes	yes	yes	-
WARCs	yes	-	yes	yes	yes	-

including professors, alumni, and fellow students, we were able to collect the necessary data using this method.

```
<script type="text/javascript">
   window._sharedData = {...
  "entry_data": { "ProfilePage": [{ "user": {
         "username": "blakelively",
         "follows": {...},
         "media": {"nodes": [{
              "code": "BGltI0Dx4EP",
              "date": 1465809056,
              "dimensions": {...},
              "comments": {...},
              "caption": "♥",
              "likes": {...},
              "owner": {"id": "1437529575"},
              "thumbnail_src": [...]
              "is_video": false,
              "id": "1271620987548827919",
              "display_src": [...]},
            {"code": "BGlshRlR4C9", [...]},
            ...
            {"code": "BGFRrDwR4OE", [...]}]},
            ...
         "followed_by_viewer": false,
         "is_verified": true,
         "external_url": [...]}}]},
  "config": {...},
  "environment_switcher_visible_server_guess": true};
</script>
```

Fig. 43. The script tag in the example HTML for the URI-M `https://web.archive.org/web/20160613225448/https://www.instagram.com/blakelively/` illustrating where the post data in account page exists

```
WARC/1.0
WARC-Date: 2019-10-25T02:36:18Z
WARC-Type: response
WARC-Record-ID: <urn:uuid:05772946-e415-4ef6-9043-3358ba583781>
WARC-Target-URI: https://www.instagram.com/p/D/
WARC-Payload-Digest: sha1:4NFVPSJRUOMUF7H2FQGX25XVNFQN5V7I
WARC-Block-Digest: sha1:LEXYO6DUDOETMMZHSPNG6SQVSZASIU22
Content-Type: application/http; msgtype=response
Content-Length: 20715

b'<!DOCTYPE html>\n<html lang="en" class="no-js not-logged-in client-root
   ">\n <head><script src="//archive.org/includes/analytics.js?v=cf34f82"
   type="text/javascript"></script>\n
...
window._sharedData = {"static_root":"\\/\\/web.archive.org\\/web
   \\/20150721221710\\/https:\\/\\/instagramstatic-a.akamaihd.net\\/
   bluebar\\/b59a4f6","entry_data":{"PostPage":[{"__query_string":"?","
   media":{"code":"C","usertags":{"nodes":[{"position":{"y":0.334967315,"x
   ":0.101307191},"user":{"username":"nicole"}}]},"owner":{"username":"
   kevin","requested_by_viewer":false,"followed_by_viewer":false,"
   has_blocked_viewer":false,"profile_pic_url":"http:\\/\\/web.archive.org
   \\/web\\/20150721221710\\/https:\\/\\/instagramimages-a.akamaihd.net\\/
   profiles\\/profile_3_75sq_1325536697.jpg","full_name":"Kevin Systrom","
   blocked_by_viewer":false,"id":"3","is_private":false},"comments":{"
   count":23773,"page_info":{"has_previous_page":true,
...
</body>\n</html><!--\n FILE ARCHIVED ON 22:17:10 Jul 21, 2015 AND
   RETRIEVED FROM THE\n INTERNET ARCHIVE ON 13:09:00 May 14, 2021.\n'
```

Fig. 44. Example WARC response record demonstrating where the username can be extracted from (case 1).

```
"owner":{"id":"3","is_verified":true,"profile_pic_url":"http://web.archive
   .org/web/20191025023618/https://scontent-lax3-1.cdninstagram.com/vp/9
   f923d0cafafba02b6250b4ff709ed3c/5E43D9F4/t51.2885-19/s150x150/13732144
   _1764457777134045_549538515_a.jpg?_nc_ht=scontent-lax3-1.cdninstagram.
   com","username":"kevin","blocked_by_viewer":false,"followed_by_viewer":
   false,"full_name":"Kevin Systrom","has_blocked_viewer":false,"
   is_private":false,"is_unpublished":false,"requested_by_viewer":false}
```


Fig. 45. A different case (case 2) to find the username of the post owner in the `script` tag which includes the id of the user account.

```
with open(file, 'rb') as fh:
    for record in ArchiveIterator(fh):
        postURI = record.rec_headers.get_header('WARC-Target-URI')
        shortcode = postURI.split("/p/")[1].strip("/")
```

Fig. 46. Python code snippet demonstrating the extraction of the username from the `WARC-Target-URI` header of the WARC response

4.4.3 Understanding the Requests to Instagram Posts in IAlogs Dataset

We looked into the Instagram URI-Ms people request from Internet Archive every year from 2011 to 2021 based on our IAlogs dataset. With the limited yearly data we have (one day per each year), we checked the number of posts requested by users every year as well as how many of those posts have been previously requested. Table 7 (columns 1 and 2) shows how many posts are requested every year (for that particular day). Figure 48 is based on the first two columns of Table 7 showing the posts requested each year. The graph depicts a steady increase in the number of posts requested each year from 2013 to 2021, with a notable surge in 2021 (over twice as many posts requested in 2020). As illustrated in Table 7 (columns 3 and 4), only 0.97% (7 out of 725) of the unique Instagram posts requested in the year 2017 had been previously requested in any of the preceding years. This indicates that web archive users predominantly request new posts each year. However, we acknowledge that this conclusion has been drawn solely by examining a single

```
#Case1
pattern = '"owner":{"username":"[a-zA-Z0-9._]+"'

#Case2
pattern = '{"graphql":{"shortcode_media":{"__typename":"GraphImage","id
   ":"\d","shortcode":"%s"' % shortcode
m = re.search(pattern, payload)
match = m.group(0)
a,id_str,code = match.split(",")
id = id_str.split(":")[1].strip('"')
pattern2 = 'owner":{' + id_str + ',"is_verified":true,"profile_pic_url
   ":".*","username":"[a-zA-Z0-9._]+"'
```

Fig. 47. Regular expressions used to match and extract the shortcode

day's worth of data for each year. This would reinforce our strategy of populating the secondary index (`IG_Index`) using the client-side JS, `unxtract` as discussed in Section 4.3.2. The higher the number of new posts requested, the greater the likelihood of additional posts being included in the `IG_Index`.

4.4.4 Creation Date of Posts Requested Each Year

We also looked into the creation date of the posts requested by the web archive user, using the IAlogs dataset. For each post, using the shortcode, we extracted the creation time of the post using a method we followed in one of our prior studies [4]. We generated two multi-line ECDF (Empirical Cumulative Distribution Function) charts, one covering the period between 2013 to 2017 (Figure 49), and the other covering the period from 2018 to 2021 (Figure 50). Our decision to split the chart into two is to ensure better clarity with each year represented by a separate line. The x-axis represents the number of months prior, meaning the number of months passed relative to the datetime of the access logs. According to Figure 49, based on access log data for years 2013 to 2017, 50% of requests are for posts that were created within just a year prior to the datetime of the access logs. Similarly, for the years 2018 to 2021 (Figure 50), 50% of requests are for posts that were created within the two years prior to the datetime of the access logs. These findings indicate that the majority of web archive users request mementos of Instagram posts that are recently cre-

Table 7. IAlogs: Requested posts from 2011 to 2021 for a single day each year (1st Thursday of February)

Year	Posts	Posts Already Seen	Percentage
2011	0	0	-
2012	0	0	-
2013	44	0	0%
2014	20	0	0%
2015	108	0	0%
2016	160	0	0%
2017	725	7	0.97%
2018	911	9	0.99%
2019	937	21	2.24%
2020	1009	31	3.07%
2021	2663	80	3.01%

ated. As we divided the chart into two, by looking at the x-axis values, we can clearly observe that the logs from 2013-2015 run back to the time when posts were created around 5 years ago, while the logs from 2018-2021 run back to the time when posts were created more than 10 years ago.

4.5 SUMMARY

In this chapter, we outlined our approaches to answering our research problem of how we can enable the discovery of all of a user's archived Instagram posts. We proposed two primary approaches for addressing this issue, which we explored in detail and evaluated for implementation feasibility. The first approach involved using WARC revisit records to incorporate Instagram usernames into the `WARC-Target-URI`, while the second approach entailed building a secondary index to map user accounts with their post URLs.

We implemented each approach to demonstrate the intended functionality. In our first approach we used WARC revisit records by incorporating Instagram account usernames into the `WARC-Target-URI`. This made it possible to conduct CDX prefix searches and obtain all archived posts of a specific Instagram user. As our second approach, we proposed building a secondary index to map each Instagram user with its post shortcodes. We showed how to populate the Instagram post index gradually using two primary techniques: `unxtract.js`, a client-side JavaScript

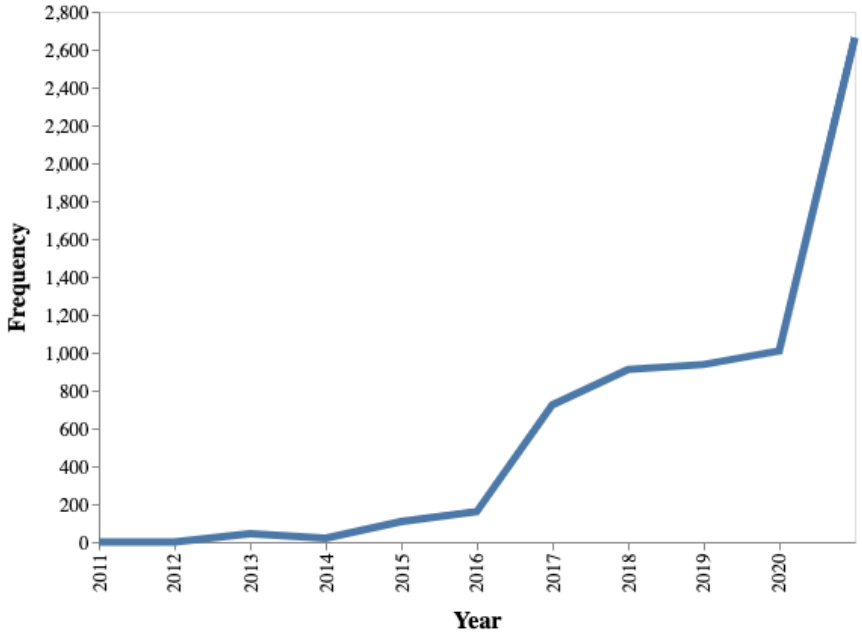

Fig. 48. The number of posts requested each year from 2011 to 2021 that shows a steady increase and a notable surge in 2021.

that extracts the username from post pages during replay, and multiple scraper scripts that extract the username from Instagram post pages and collect all the posts from Instagram account pages from both live and archived web

In order to evaluate the feasibility of these proposed approaches, we considered requirements based on factors such as storage, computational power, and other dependencies. Additionally, we discussed the challenges we encountered during the data collection process using live web scraping methods on Instagram. We also shared some interesting insights from the IAlogs dataset that we used for various implementations in our study.

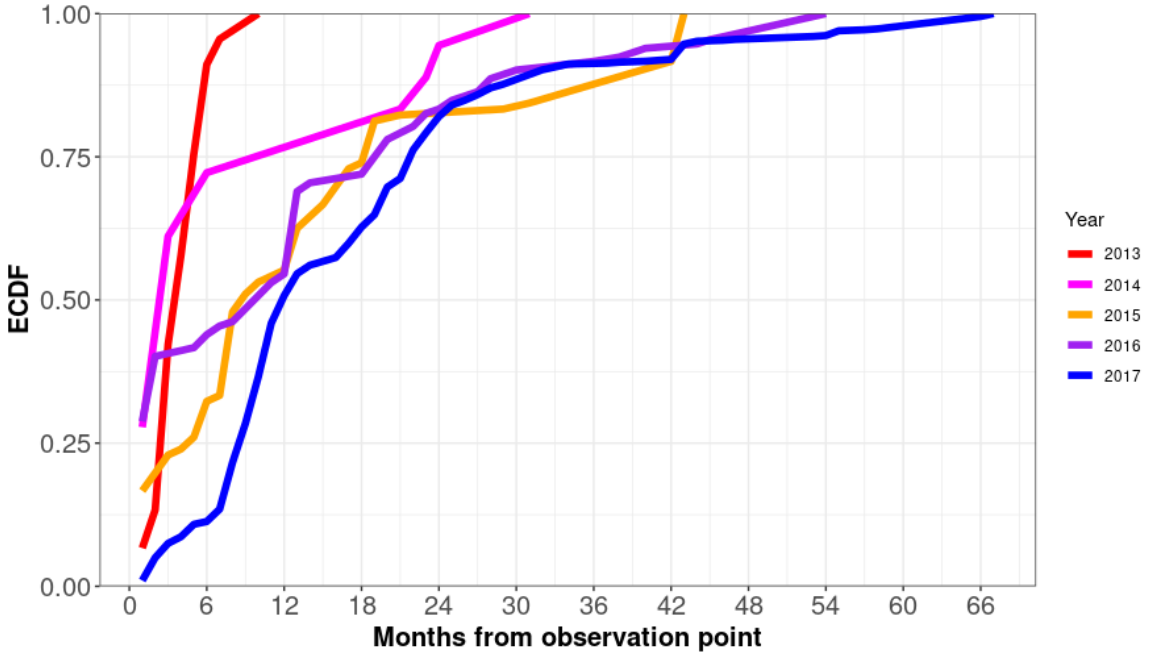

Fig. 49. Creation date of posts requested each year: 2013-2017

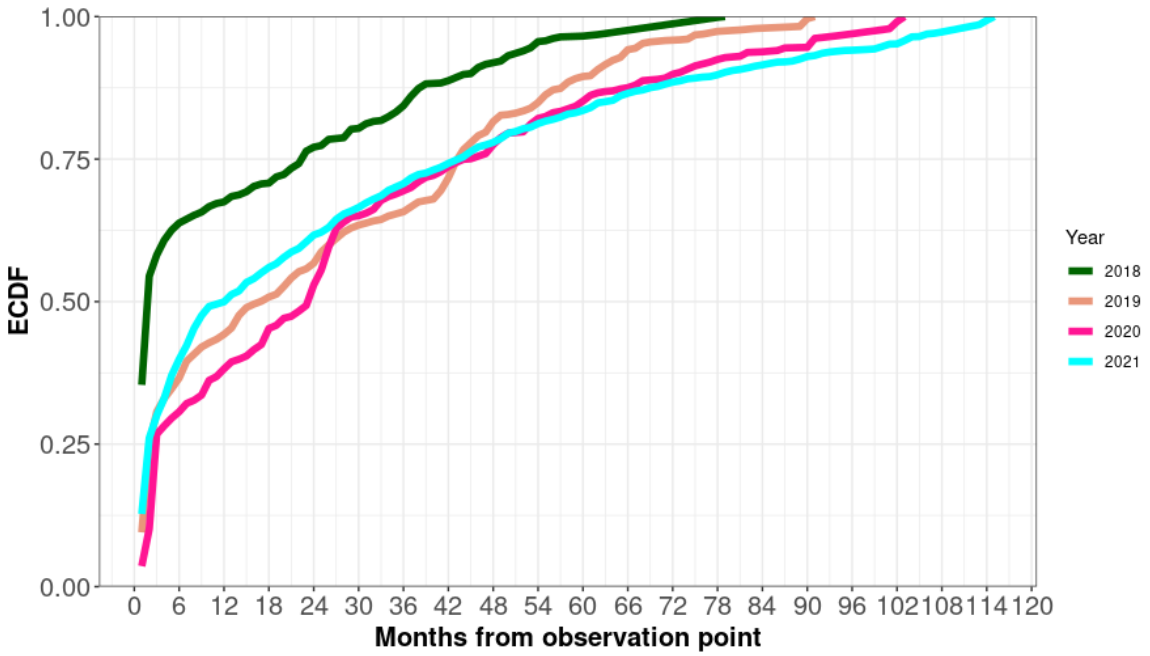

Fig. 50. Creation date of posts requested each year: 2018-2021

CHAPTER 5

FUTURE WORK

We proposed creating a secondary index to support discovering mementos for Instagram posts of a specific user as one of our approaches. For future work, it would be valuable to evaluate the time required for constructing an index of all the Instagram posts in the Internet Archive. This would pose challenges due to the fact that most of the recent Instagram mementos redirect to the Instagram login page, making it difficult to determine the number of usable mementos by merely examining CDX or TimeMap. Additionally, although we developed the index locally and offered several approaches for populating the index as proof of concept, we refrained from publicly hosting the index at this stage. Future research should examine the possible advantages and disadvantages of making the secondary index accessible to the general public, especially in terms of scalability and security. There are other deciding factors to hosting the proposed index remotely including issues such as network latency, additional hosting costs, security concerns, and potential dependencies on other third-party services. Further research could be conducted to identify methods for optimizing the index's performance when hosted on a remote server. Although we did not address ethical and legal concerns regarding the archiving and access of Instagram posts, such as privacy and copyright issues, we acknowledge their significance and potential for future research. While we have provided a summary of the requirements for various approaches, including archive support, processing time, storage, and user preference, we have not quantified the precise amount of each requirement needed, nor have we determined the specific factors influencing each requirement. This area could be the subject of future research. We understand that the CDX format is a commonly used format in web archiving and is supported by several web archiving organizations, including the Internet Archive, the UK web archive,[1] and the Library of Congress.[2] However, we did not conduct an investigation of CDX server API support, particularly with respect to prefix search, among other web archiving entities beyond the Wayback Machine at the Internet Archive. Our objective is to expand our research by exploring additional approaches for archives that do not support prefix search. Nevertheless, our secondary index approach remains a reliable option for accessing a user's Instagram posts in web archives.

[1] https://www.webarchive.org.uk/ukwa/

[2] https://www.loc.gov/

CHAPTER 6

CONCLUSION

Social media has gained immense popularity and is now a crucial communication channel for people of all ages. It is a key source of information on current events and cultural and historical developments of our time. We discussed how Instagram is one of the rising social media platforms and its crucial role in shaping our current social and cultural landscape. However, we also noted the lack of academic research dedicated to Instagram when compared to its counterparts, such as Twitter and Facebook. This lack of academic research indicates a gap in our understanding of the platform's full potential, as well as its impact on society. We pointed out some of the components of the limited academic research on Instagram including the lack of API support for academic research, the lack of a built-in sharing feature, its focus on posting only images (not text), and how the platform's involvement in various operations are not always immediately apparent. We highlighted the significance of preserving Instagram content as much as other web resources. We explored the challenges of accessing Instagram content, specifically posts belonging to a particular user, through web archives. We drew a comparison with the process of finding all the posts of a specific user on Facebook or Twitter in web archives to highlight this issue. We studied Instagram's URL structure, which is the underlying source of the issue.

During our investigation, we explored various methods to support the discovery of Instagram posts owned by a particular user. We identified two main approaches: 1) using revisit records to incorporate Instagram usernames into the WARC record header, WARC-Target-URI and 2) building a secondary index to map user accounts to their post URLs. To implement the Revisit Record approach, revisit records are created with the `WARC-Target-URI` containing the post URI modified to include the username and `WARC-Refers-To-Target-URI` referencing the original post URI. Once the revisit records are created and indexed by the Wayback Machine, a mapping is established between the URLs of the original Instagram post (`instagram.com/p/{shortcode}`) and the post URL that includes the username (`instagram.com/{username}/p/{shortcode}`). This enables both URI-Rs to point to the same memento of the post. Due to the URL structure in the revisit record, the prefix search would automatically work on these records as well. The Secondary Index approach involves building a separate external index that is designed to enable users to retrieve all post URLs for a specific Instagram user. This index includes a mapping of shortcodes to the corresponding usernames. By using these shortcodes, the users can construct the post URLs and query the Internet Archive for each individual post separately. We also explored various meth-

ods that can be used to populate the index. The first method involves using our proposed JavaScript file (unxtract.js) in conjunction with the well-known wombat.js designed to extract the usernames from Instagram posts during the replay on the client-side. Alternatively, we demonstrated the use of various scraper scripts to extract the username from scraped posts and account pages from both the live and archived web, as well as from WARC files for Instagram posts.

We have successfully implemented all of the mentioned approaches locally and demonstrated their functionality in supporting the discovery of archived Instagram posts belonging to a particular user. We conducted testing using multiple accounts and posts to verify the effectiveness of these methods. We analyzed the server access logs of the Internet Archive from 2011 to 2021 (one day per year), which revealed an upward trend in the number of requests for the Instagram domain over the years, especially until 2020. However, it is worth noting that the proportion of Instagram requests remained less than 0.2% each year. Our findings also show a consistent increase in the number of posts requested each year from 2013 to 2021, with a surge in 2021, wherein the number of requested posts is over twice that of 2020. Further analysis of access log data for the years 2013 to 2017 revealed that 50% of the requests were for posts created within one year prior to the datetime of the access logs. Similarly, for the years 2018 to 2021, 50% of requests were for posts created within two years prior to the datetime of the access logs. These results indicate that the majority of web archive users request mementos of Instagram posts that are created recently. In addition to our small-scale demo implementations and exploratory evaluation, we discussed the advantages and disadvantages of the proposed approaches. This would enable web archivists to make informed decisions on which approach to adopt based on practicality and unique requirements for their archive.

REFERENCES

[1] Christina Newberry. 42 Facebook Statistics Marketers Need to Know in 2023. https://blog.hootsuite.com/facebook-statistics/ (2023).

[2] Ash Turner. How Many Users Does Twitter Have? (Jan 2023). https://www.bankmycell.com/blog/how-many-users-does-twitter-have (2023).

[3] Stuart Dredge. Instagram now has more than 2bn monthly active users. https://musically.com/2022/10/28/instagram-now-has-more-than-2bn-monthly-active-users/ (2022).

[4] Himarsha R. Jayanetti. Creation Time and Published Time Are Not the Same: Estimating the Instagram Epoch. https://ws-dl.blogspot.com/2021/02/2021-02-20-creation-time-and-published.html (2021).

[5] Twitter. Twitter API. https://developer.twitter.com/en/products/twitter-api (2023).

[6] Meta. Instagram Graph API. https://developers.facebook.com/docs/instagram-api (2023).

[7] Meta. Instagram Basic Display API. https://developers.facebook.com/docs/instagram-basic-display-api (2023).

[8] Stewart, L. G., Arif, A. & Starbird, K. Examining Trolls and Polarization with a Retweet Network. In *Proceedings of the ACM WSDM Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*, vol. 70 (2018). https://faculty.washington.edu/kstarbi/examining-trolls-polarization.pdf.

[9] Starbird, K., Arif, A. & Wilson, T. Disinformation as Collaborative Work: Surfacing the Participatory Nature of Strategic Information Operations. vol. 3 (2019). https://doi.org/10.1145/3359229.

[10] Starbird, K. *et al.* Ecosystem or Echo-System? Exploring Content Sharing across Alternative Media Domains. In *Proceedings of the International AAAI Conference on Web and Social Media* (2018). https://doi.org/10.1609/icwsm.v12i1.15009.

[11] Garg, K., Jayanetti, H. R., Alam, S., Weigle, M. C. & Nelson, M. L. Replaying archived twitter: When your bird is broken, will it bring you down? In *Proceedings of the 2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 160–169 (2021).

[12] Sesagiri Raamkumar, A., Tan, S. G. & Wee, H. L. Measuring the outreach efforts of public health authorities and the public response on facebook during the COVID-19 pandemic in early 2020: cross-country comparison. *Journal of medical Internet research* **22**, e19334 (2020).

[13] Renee DiResta, Kris Shaffer, Becky Ruppel, David Sullivan, Robert Matney, Ryan Fox, Jonathan Albright, Ben Johnson. The Tactics & Tropes of the Internet Research Agency. https://www.intelligence.senate.gov/sites/default/files/documents/NewKnowledge-Disinformation-Report-Whitepaper.pdf (2018).

[14] Bragg, H. & Weigle, M. C. Discovering the Traces of Disinformation on Instagram in the Internet Archive. Tech. Rep. arXiv:2301.09188 (2023). URL https://arxiv.org/abs/2301.09188. https://doi.org/10.48550/ARXIV.2301.09188.

[15] Bragg, H., Jayanetti, H. R., Nelson, M. L. & Weigle, M. C. Less than 4% of archived Instagram account pages for the Disinformation Dozen are replayable. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (2023).

[16] BBC News. Covid: Twitter suspends Naomi Wolf after tweeting anti-vaccine misinformation. https://www.bbc.com/news/world-us-canada-57374241 (2021).

[17] Berners-Lee, T. & Connolly, D. Hypertext Markup Language - 2.0 - RFC 1866. https://www.ietf.org/rfc/rfc1866.txt (1995).

[18] Fielding, R. T. & Reschke, J. F. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing - RFC 7230. https://tools.ietf.org/html/rfc7230 (2014).

[19] Berners-Lee, T., Fielding, R. T. & Masinter, L. Uniform Resource Identifier (URI): Generic Syntax - RFC 3986. https://www.rfc-editor.org/rfc/rfc3986 (2005).

[20] W3C. URI/Resource Relationships. https://www.w3.org/TR/webarch/#id-resources.

[21] W3C. Architecture of the World Wide Web, Volume One. https://www.w3.org/TR/webarch/#uri-opacity (2004).

[22] Alam, S., Cartledge, C. L. & Nelson, M. L. Support for Various HTTP Methods on the Web. Tech. Rep. arXiv:1405.2330, Old Dominion University (2014). https://doi.org/10.48550/arXiv.1405.2330.

[23] Tim Berners-Lee. Cool URIs don't change. https://www.w3.org/Provider/Style/URI (1998).

[24] Evelyn M. Rusli. Facebook Buys Instagram for $1 Billion. https://archive.nytimes.com/dealbook.nytimes.com/2012/04/09/facebook-buys-instagram-for-1-billion/ (2012).

[25] W3C. Cool URIs for the Semantic Web. https://www.w3.org/TR/cooluris/#cooluris (2008).

[26] IPFS. IPFS comparisons. https://docs.ipfs.tech/concepts/comparisons/ (2015).

[27] MAGNET-URI Project. https://magnet-uri.sourceforge.net.

[28] Watanabe, T., Shioji, E., Akiyama, M. & Mori, T. Melting Pot of Origins: Compromising the Intermediary Web Services that Rehost Websites. In *Network and Distributed System Security Symposium* (2020). https://doi.org/10.14722/ndss.2020.24140.

[29] Van de Sompel, H., Nelson, M. L. & Sanderson, R. HTTP Framework for Time-Based Access to Resource States - Memento - RFC 7089. http://tools.ietf.org/html/rfc7089 (2013).

[30] International Internet Preservation Consortium. The WARC Format 1.1. https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/.

[31] Internet Archive: CDX file format. http://archive.org/web/researcher/cdx_file_format.php (2003).

[32] Berlin, J., Kelly, M., Nelson, M. L. & Weigle, M. C. To re-experience the web: A framework for the transformation and replay of archived web pages. *ACM Transactions on the Web* to appear (2023). https://doi.org/10.1145/3589206.

[33] Ruby Seavey. Elon Musk's Twitter continues to favor right-wing content, reinstating dozens of accounts with millions of combined followers. https://www.mediamatters.org/e

lon-musk/elon-musks-twitter-continues-favor-right-wing-content-reinstating-dozens-accounts (2022).

[34] Associated Press. Kanye West's Twitter, Instagram locked over offensive posts. https://www.politico.com/news/2022/10/10/kanye-wests-twitter-instagram-locked-over-offensive-posts-00061089 (2022).

[35] Jak Connor. Kanye West returns to Instagram with one post, immediately gets banned again. https://www.tweaktown.com/news/89274/kanye-west-returns-to-instagram-with-one-post-immediately-gets-banned-again/index.html (2022).

[36] Emma Wilkes. Kanye West banned from Instagram again after sharing clip of new song. https://www.nme.com/news/music/kanye-west-banned-from-instagram-again-after-sharing-clip-of-new-song-3364340 (2022).

[37] Kanishka Singh. Twitter suspends account of white supremacist Nick Fuentes a day after restoration. https://www.reuters.com/world/us/twitter-suspends-account-white-supremacist-nick-fuentes-day-after-restoration-2023-01-25/ (2023).

[38] Braghieri, L., Levy, R. & Makarin, A. Social media and mental health. *American Economic Review* **112**, 3660–3693 (2022). http://doi.org/10.1257/aer.20211218.

[39] Valkenburg, P. M., Meier, A. & Beyens, I. Social media use and its impact on adolescent mental health: An umbrella review of the evidence. *Current Opinion in Psychology* **44**, 58–68 (2022). https://doi.org/10.1016/j.copsyc.2021.08.017.

[40] OâĂŹreilly, M. Social media and adolescent mental health: the good, the bad and the ugly. *Journal of Mental Health* **29**, 200–206 (2020). https://doi.org/10.1080/09638237.2020.1714007.

[41] Rainie, L. *et al.* Social media and political engagement. *Pew Internet & American Life Project* **19**, 2–13 (2012). https://www.pewresearch.org/internet/2012/10/19/social-media-and-political-engagement/.

[42] Kubin, E. & von Sikorski, C. The role of (social) media in political polarization: a systematic review. *Annals of the International Communication Association* **45**, 188–206 (2021). https://doi.org/10.1080/23808985.2021.1976070.

[43] Waeterloos, C., Walrave, M. & Ponnet, K. Designing and validating the social media political participation scale: An instrument to measure political participation on social media. *Technology in Society* **64**, 101493 (2021). https://doi.org/10.1016/j.techsoc.2020.101493.

[44] Chung, W. Social media analytics: Security and privacy issues. *Journal of Information Privacy and Security* **12**, 105–106 (2016).

[45] Kushwah, V. R. S. & Verma, K. Security and privacy challenges for big data on social media. *Big data analytics in cognitive social media and literary texts: Theory and praxis* 267–285 (2021).

[46] Kumar, R., Kumar, P. & Kumar, V. Design and implementation of privacy and security system in social media. *International Journal of Advanced Networking and Applications* **13**, 5081–5088 (2022).

[47] Oppliger, R. Security and privacy in an online world. *Computer* **44**, 21–22 (2011).

[48] Mason, A. N., Narcum, J. & Mason, K. Social media marketing gains importance after COVID-19. *Cogent Business & Management* **8**, 1870797 (2021). https://doi.org/10.1080/23311975.2020.1870797.

[49] Syaifullah, J., Syaifudin, M., Sukendar, M. U. & Junaedi, J. Social media marketing and business performance of msmes during the COVID-19 pandemic. *The Journal of Asian Finance, Economics and Business* **8**, 523–531 (2021). https://doi.org/10.13106/jafeb.2021.vol8.no2.0523.

[50] Moslehpour, M., Ismail, T., Purba, B. & Wong, W.-K. What makes GO-JEK go in Indonesia? The influences of social media marketing activities on purchase intention. *Journal of Theoretical and Applied Electronic Commerce Research* **17**, 89–103 (2021). https://doi.org/10.3390/jtaer17010005.

[51] Chatterjee, S. & Kar, A. K. Why do small and medium enterprises use social media marketing and what is the impact: Empirical insights from India. *International Journal of Information Management* **53**, 102103 (2020). https://doi.org/10.1016/j.ijinfomgt.2020.102103.

[52] Abbas, J. *et al.* The effects of corporate social responsibility practices and environmental factors through a moderating role of social media marketing on sustainable performance of business firms. *Sustainability* **11**, 3434 (2019).

[53] Shu, K., Mahudeswaran, D., Wang, S., Lee, D. & Liu, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data* **8**, 171–188 (2020).

[54] Shu, K., Wang, S., Lee, D. & Liu, H. *Disinformation, misinformation, and fake news in social media* (Springer, 2020).

[55] Rocha, Y. M. *et al.* The impact of fake news on social media and its influence on health during the COVID-19 pandemic: A systematic review. *Journal of Public Health* 1–10 (2021).

[56] Dan, V. *et al.* Visual mis-and disinformation, social media, and democracy. *Journalism & Mass Communication Quarterly* **98**, 641–664 (2021).

[57] Chadwick, A. & Vaccari, C. News sharing on UK social media: Misinformation, disinformation, and correction (2019).

[58] Lomborg, S. & Bechmann, A. Using APIs for data collection on social media. *The Information Society* **30**, 256–265 (2014). https://doi.org/10.1080/01972243.2014.915276.

[59] Hogan, B. Social media giveth, social media taketh away: Facebook, friendships, and APIs. *International Journal of Communication* 592–611 (2016). URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3084159.

[60] Bruns, A. After the 'APIcalypse': social media platforms and their fight against critical scholarly research. *Information, Communication & Society* **22**, 1544–1566 (2019). https://doi.org/10.1080/1369118X.2019.1637447.

[61] Hartman, R. & Simova, T. How changing API terms changed Instagram's domain? A bibliometric analysis. In *2021 7th International Conference on Computer Technology Applications*, 73–79 (2021).

[62] Eva Rtology. February 13th Twitter will decrease by 80%? no more FREE API. https://medium.com/mlearning-ai/february-13th-twitter-will-decrease-by-80-no-more-free-api-72e7835d6806 (2023).

[63] Rowe, F., Mahony, M., Graells-Garrido, E., Rango, M. & Sievers, N. Using twitter to track immigration sentiment during early stages of the COVID-19 pandemic. *Data & Policy* **3**, e36 (2021). https://doi.org/10.1017/dap.2021.38.

[64] Manguri, K. H., Ramadhan, R. N. & Amin, P. R. M. Twitter sentiment analysis on worldwide COVID-19 outbreaks. *Kurdistan Journal of Applied Research* 54–65 (2020).

[65] Wang, L., Niu, J. & Yu, S. Sentidiff: combining textual information and sentiment diffusion patterns for twitter sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* **32**, 2026–2039 (2019).

[66] Keller, F. B., Schoch, D., Stier, S. & Yang, J. Political astroturfing on Twitter: How to coordinate a disinformation campaign. *Political communication* **37**, 256–280 (2020).

[67] Manfredi-Sánchez, J.-L., Amado-Suárez, A. & Waisbord, S. Presidential Twitter in the face of COVID-19: Between populism and pop politics. *Comunicar: Media Education Research Journal* **29**, 79–90 (2021).

[68] Theocharis, Y., Barberá, P., Fazekas, Z. & Popa, S. A. The dynamics of political incivility on Twitter. *Sage Open* **10** (2020).

[69] Barrie, C. & Ho, J. C.-t. academictwitteR: an r package to access the Twitter academic research product track v2 API endpoint. *Journal of Open Source Software* **6**, 3272 (2021).

[70] Müller, M., Salathé, M. & Kummervold, P. E. COVID-Twitter-BERT: A natural language processing model to analyse COVID-19 content on Twitter. Tech. Rep. arXiv:2005.07503 (2020). URL https://arxiv.org/abs/2005.07503.

[71] Chen, E. & Ferrara, E. Tweets in time of conflict: A public dataset tracking the Twitter discourse on the war between Ukraine and Russia. Tech. Rep. arXiv:2203.07488 (2022). https://doi.org/10.48550/ARXIV.2203.07488.

[72] Pohl, J., Seiler, M. V., Assenmacher, D. & Grimme, C. A Twitter streaming dataset collected before and after the onset of the war between Russia and Ukraine in 2022. *Available at SSRN* (2022).

[73] Sazzed, S. The dynamics of Ukraine-Russian conflict through the lens of demographically diverse Twitter data. In *2022 IEEE International Conference on Big Data (Big Data)*, 6018–6024 (IEEE, 2022).

[74] Pfeffer, J. *et al.* This sample seems to be good enough! assessing coverage and temporal reliability of Twitter's academic API. Tech. Rep. arXiv:2204.02290 (2022). https://doi.org/10.48550/ARXIV.2204.02290.

[75] Chen, K., Duan, Z. & Yang, S. Twitter as research data: Tools, costs, skill sets, and lessons learned. *Politics and the Life Sciences* **41**, 114âĂŞ130 (2022). https://doi.org/10.1017/pls.2021.19.

[76] Vázquez-Herrero, J., Direito-Rebollal, S. & López-García, X. Ephemeral journalism: News distribution through Instagram stories. *Social media+ society* **5**, 2056305119888657 (2019).

[77] Sharma, K. A. & Naresh, S. Setting narrative through Instagram posts: A study of BBC's reportage on Afghanistan. *Arab Studies Quarterly* **44**, 84–96 (2022). URL https://www.jstor.org/stable/48675927.

[78] Al-Rawi, A., Al-Musalli, A. & Fakida, A. News values on Instagram: A comparative study of international news. *Journalism and Media* **2**, 305–320 (2021). https://doi.org/10.3390/journalmedia2020018.

[79] Hu, Y., Manikonda, L. & Kambhampati, S. What we Instagram: A first analysis of Instagram photo content and user types. In *Proceedings of the international AAAI conference on web and social media*, vol. 8, 595–598 (2014).

[80] Doney, J., Wikle, O. & Martinez, J. Likes, comments, views. *Information Technology and Libraries* **39** (2020).

[81] Wilkinson, J. Accessible, dynamic web content using Instagram. *Information Technology and Libraries* **37**, 19–26 (2018).

[82] Max Bamber. Introduction to using Instagram to promote archive collections. https://blog.townswebarchiving.com/2016/11/introduction-to-using-instagram-to-promote-archive-collections (2016).

[83] BBC. Twitter: Five ways Elon Musk has changed the platform for users. https://www.bbc.com/news/technology-64289251 (2023).

[84] Viktor Hendelmann. What Happened To Friendster? 4 Reasons Why It Failed. https://productmint.com/what-happened-to-friendster/ (2023).

[85] Google Support. Shutting down Google+ for consumer (personal) accounts on April 2, 2019. https://support.google.com/googlecurrents/answer/9195133 (2019).

[86] Xing, S. & Paris, B.-P. Measuring the size of the Internet via importance sampling. *IEEE Journal on Selected Areas in Communications* **21**, 922–933 (2003). https://doi.org/10.1109/JSAC.2003.814510.

[87] Van den Bosch, A., Bogers, T. & De Kunder, M. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics* **107**, 839–856 (2016). https://doi.org/10.1007/s11192-016-1863-z.

[88] Parks, M. What will we study when the Internet disappears? *Journal of Computer-Mediated Communication* **14**, 724–729 (2009). https://doi.org/10.1111/j.1083-6101.2009.01462.x.

[89] Ainsworth, S. G., Alsum, A., SalahEldeen, H., Weigle, M. C. & Nelson, M. L. How much of the web is archived? In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, 133–136 (Association for Computing Machinery, New York, NY, USA, 2011). https://doi.org/10.1145/1998076.1998100.

[90] Milligan, I. Lost in the infinite archive: The promise and pitfalls of web archives. *International Journal of Humanities and Arts Computing* **10**, 78–94 (2016).

[91] Justin Littman. Web archiving and/or/vs social media API archiving. https://gwu-libraries.github.io/sfm-ui/posts/2017-12-13-web-social-media-archiving (2017).

[92] Littman, J. *et al.* API-based social media collecting as a form of web archiving. *International Journal on Digital Libraries* **19**, 1–18 (2018). https://doi.org/10.1007/s00799-016-0201-7.

[93] Wang, X. *Large Web Archive Collection Infrastructure and Services*. Ph.D. thesis, Virginia Tech (2023).

[94] Speaker, S. L. & Moffatt, C. The national library of medicine global health events web archive, coronavirus disease (COVID-19) pandemic collecting. *Journal of the Medical Library Association: JMLA* **108**, 656 (2020).

[95] Sara Day Thomson. Catching an Avalanche with a Teaspoon: the Global Challenge of Web Archiving the Coronavirus Pandemic. https://www.dpconline.org/blog/series-wa-coronavirus-sdthomson-2 (2020).

[96] Sara Day Thomson. If These WARCs Could Talk: Learning from Archived Web & Social Media COVID-19 Collections. https://www.dpconline.org/blog/series-wa-coronavirus-sdthomson-4 (2020).

[97] John OBrien III. yourTwapperKeeper. https://github.com/540co/yourTwapperKeeper.

[98] Zhang, Z. & Nasraoui, O. Profile-based focused crawler for social media-sharing websites. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, vol. 1, 317–324 (2008). https://doi.org/10.1109/ICTAI.2008.119.

[99] Himarsha R. Jayanetti. Twitter rewrites your URLs, but assumes you'll never rewrite theirs: more problems replaying archived Twitter. https://ws-dl.blogspot.com/2021/01/2020-01-22-twitter-rewrites-your-urls.html (2021).

[100] Himarsha R. Jayanetti . New Twitter UI: Replaying Archived Twitter Pages That Never Existed. https://ws-dl.blogspot.com/2020/11/2020-11-04-new-twitter-ui-replaying.html (2020).

[101] Brunelle, J., Kelly, M., SalahEldeen, H., Weigle, M. C. & Nelson, M. L. Not all mementos are created equal: Measuring the impact of missing resources. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 321–330 (London, 2014). https://doi.org/10.1109/JCDL.2014.6970187.

[102] Brunelle, J., Kelly, M., SalahEldeen, H., Weigle, M. C. & Nelson, M. L. Not all mementos are created equal: Measuring the impact of missing resources. *International Journal of Digital Libraries (IJDL)* **16**, 283–301 (2015). https://doi.org/10.1007/s00799-015-0150-6.

[103] Kelly, M., Nelson, M. L. & Weigle, M. C. The archival acid test: Evaluating archive performance on advanced html and javascript. In *IEEE/ACM Joint Conference on Digital Libraries*, 25–28 (2014). https://doi.org/10.1109/JCDL.2014.6970146.

[104] Goel, A., Zhu, J. & Madhyastha, H. V. Making links on your web pages last longer than you. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, HotNets '22, 145–151 (Association for Computing Machinery, New York, NY, USA, 2022). https://doi.org/10.1145/3563766.3564103.

[105] SalahEldeen, H. M. & Nelson, M. L. Losing my revolution: How many resources shared on social media have been lost? In *Theory and Practice of Digital Libraries: Second International Conference, TPDL 2012, Paphos, Cyprus, September 23-27, 2012. Proceedings 2*, 125–137 (Springer, 2012).

[106] Van de Sompel, H., Klein, M. & Shankar, H. Towards robust hyperlinks for web-based scholarly communication. In *International Conference on Intelligent Computer Mathematics* (2014).

[107] Jones, S. M., Klein, M. & Van de Sompel, H. Robustifying links to combat reference rot. *Code4Lib Journal* **50** (2021).

[108] Klein, M. *et al.* Scholarly context not found: One in five articles suffers from reference rot. *PLOS ONE* **9**, 1–39 (2014). https://doi.org/10.1371/journal.pone.0115253.

[109] Klein, M., Shankar, H. & Van de Sompel, H. Robust links in scholarly communication. In *JCDL'18: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, 357–358 (Association for Computing Machinery, New York, NY, USA, 2018). https://doi.org/10.1145/3197026.3203885.

[110] Jones, S. M. *et al.* Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLoS ONE* **11** (2016). https://doi.org/10.1371/journal.pone.0167475.

[111] Sharma, S. "how tweet it is!": Have Twitter archives been left in the dark? *Illinois Journal of Law, Technology & Policy* 49–78 (2019).

[112] O'Halloran, C. *Uncomfortable Untruths in the Archive: The Irish Slaves Meme and the Creation of Controversial Social Media Archives*. Ph.D. thesis, Leiden University (2019).

[113] Duncan, S. & Blumenthal, K.-R. A collaborative model for web archiving ephemeral art resources at the New York Art Resources Consortium (NYARC). *Art Libraries Journal* **41**, 116–126 (2016). https://doi.org/10.1017/alj.2016.12.

[114] Bainotti, L., Caliandro, A. & Gandini, A. From archive cultures to ephemeral content, and back: Studying Instagram Stories with digital methods. *New Media & Society* **23**, 3656–3676 (2021). https://doi.org/10.1177/1461444820960071.

[115] MacDowall, L. J. & de Souza, P. "I'd double tap that!!": street art, graffiti, and Instagram research. *Media, culture & society* **40**, 3–22 (2018).

[116] IIPC. Get Involved in Web Archiving Street Art. https://netpreserveblog.wordpress.com/2022/12/13/get-involved-in-web-archiving-street-art/ (2022).

[117] Peter Chan. Archiving Instagram posts. https://library.stanford.edu/blogs/stanford-libraries-blog/2021/09/archiving-instagram-posts (2022).

[118] CCDH. Malgorithm - Fix Instagram. https://counterhate.com/research/malgorithm-fix-instagram/ (2022).

[119] Center for Countering Digital Hate (CCDH). The Disinformation Dozen: Why Platforms must act on twelve leading online anti-vaxxers. https://counterhate.com/research/the-disinformation-dozen/ (2021).

[120] Jaffe, E. & Kirkpatrick, S. Architecture of the Internet Archive. In *Proceedings of SYSTOR: The Israeli Experimental Systems Conference*, 1–10 (2009).

[121] Stack, M. Searching web archive collections. *Open Source Web Information Retrieval* 51 (2005).

[122] Stack, M. Full text search of web archive collections. *Proceedings of IWAW: International Web Archiving Workshop* (2006).

[123] Cruz, D. & Gomes, D. Adapting search user interfaces to web archives. In *Proc. of the 10th International Conference on Preservation of Digital Objects (iPres)*, vol. 17 (2013).

[124] Costa, M. Full-text and URL search over web archives. In *The past web: exploring web archives*, 71–84 (Springer, 2021).

[125] Abrams, S. *et al.* Sowing the seeds for more usable web archives: a usability study of archive-it. *The American Archivist* **82**, 440–469 (2019).

[126] Gomes, D., Miranda, J. & Costa, M. A survey on web archiving initiatives. In *Research and Advanced Technology for Digital Libraries: International Conference on Theory and Practice of Digital Libraries, TPDL 2011, Berlin, Germany, September 26-28, 2011. Proceedings 1*, 408–420 (Springer, 2011).

[127] AlSum, A., Nelson, M. L., Sanderson, R. & Van de Sompel, H. Archival HTTP redirection retrieval policies. In *Proceedings of the 22nd International Conference on World Wide Web*, 1051–1058 (2013).

[128] Costa, M., Gomes, D., Couto, F. & Silva, M. A survey of web archive search architectures. In *Proceedings of the 22nd international conference on world wide web*, 1045–1050 (2013).

[129] Cheng, M., Wu, Y., Zhou, X., Li, J. & Zhang, L. Efficient web archive searching. Virginia Tech, https://vtechworks.lib.vt.edu/handle/10919/98241 (2020).

[130] Kanhabua, N. *et al.* How to search the Internet Archive without indexing it. In *Research and Advanced Technology for Digital Libraries: 20th International Conference on Theory and Practice of Digital Libraries, TPDL 2016, Hannover, Germany, September 5–9, 2016, Proceedings 20*, 147–160 (Springer, 2016).

[131] Vo, K. D., Tran, T., Nguyen, T. N., Zhu, X. & Nejdl, W. Can we find documents in web archives without knowing their contents? In *Proceedings of the 8th ACM Conference on Web Science*, 173–182 (2016).

[132] Nauman Siddique. Searching Web Archives for Unattributed Deleted Tweets From Politwoops. https://ws-dl.blogspot.com/2019/08/2019-08-03-searching-web-archives-for.html (2022).

[133] Summers, E. Trump's Tweets. https://inkdroid.org/2021/01/21/trumps-tweets/ (2021).

[134] Holzmann, H., Goel, V. & Anand, A. Archivespark: Efficient web archive access, extraction and derivation. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, 83–92 (2016).

[135] Kelly, M. *et al.* Impact of URI canonicalization on memento count. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 1–2 (IEEE, 2017).

[136] Alam, S. & Nelson, M. L. MemGator - A Portable Concurrent Memento Aggregator. In *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '16, 243–244 (ACM, New York, NY, USA, 2016). http://doi.org/10.1145/2910896.2925452.

[137] Alam, S. *et al.* Web archive profiling through CDX summarization. *International Journal on Digital Libraries* **17**, 223–238 (2016).

[138] Yves Maurer. Investigate holdings of web archives through summaries: cdx-summarize. https://netpreserveblog.wordpress.com/2022/08/10/investigate-holdings-of-web-archives-through-summaries-cdx-summarize/ (2022).

[139] Alam, S. & Graham, M. CDX summary: Web archival collection insights. In *Linking Theory and Practice of Digital Libraries: 26th International Conference on Theory and Practice of*

Digital Libraries, TPDL 2022, Padua, Italy, September 20–23, 2022, Proceedings, 297–305 (Springer, 2022).

[140] Jayanetti, H. R., Garg, K., Alam, S., Nelson, M. L. & Weigle, M. C. Robots still outnumber humans in web archives, but less than before. In *Proceedings of the Theory and Practice of Digital Libraries Conference (TPDL)* (2022). https://doi.org/10.1007/978-3-031-16802-4_19.

[141] Michael L. Nelson . Not Your ParentsâĂŹ Web: Scope, Segmentation, Stability, Resilience, and Persistence. https://ws-dl.blogspot.com/2021/10/2021-10-20-not-your-parents-web-scope.html (2021).

[142] IIPC. The WARC Format 1.1 - âĂŸrevisitâĂŹ. https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1//#revisit.

[143] Ilya Kreymer. WARCIO. https://github.com/webrecorder/warcio.

[144] Ilya Kreymer. Webrecorder pywb 2.7.2. https://github.com/webrecorder/pywb (2014).

[145] Webrecorder. CDXJ Server API. https://pywb.readthedocs.io/en/latest/manual/cdxserver_api.html (2014).

[146] Himarsha R. Jayanetti. How well is Instagram archived? https://ws-dl.blogspot.com/2020/11/2020-11-04-how-well-is-instagram.html (2020).

[147] Loesing, K., Murdoch, S. J. & Dingledine, R. A case study on measuring statistical data in the Tor anonymity network. In *Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010)*, LNCS (Springer, 2010). https://doi.org/10.1007/978-3-642-14992-4_19.

APPENDIX A

THE CLIENT-SIDE JAVASCRIPT IMPLEMENTATION - unxtract.js

This JavaScript code is employed for extracting the `script` tag that contains the Instagram account username of the post owner. After extracting the value of the `script` tag, the JavaScript library will forward it to the `add_un_idx` endpoint of the API. At this endpoint, the username extraction process will take place before the user is added to the index. This is discussed further in Section 4.3.2.

```
1  window.addEventListener("load", function() {
2      let url = document.getElementById('replay_iframe').src;
3      let iframe =
       ↪  document.getElementById('replay_iframe').contentDocument;
4      const scripts = iframe.getElementsByTagName("script");
5      const username = scripts[17].innerHTML;
6      console.log(url, username);
7
8      var xhr = new XMLHttpRequest();
9
10     xhr.onreadystatechange = (e) => {
11         if (xhr.readyState !== 4) {
12             return;
13         }
14
15         if (xhr.status === 200) {
16             console.log('SUCCESS', xhr.responseText);
17         } else {
18             console.warn('request_error');
19         }
20
21         var headers = xhr.getAllResponseHeaders().toLowerCase();
22         console.log(headers)
23     };
24     xhr.open("POST", "http://localhost:5000/add_un_idx", true);
```

```
25      xhr.setRequestHeader('Content-Type', 'application/json');
26      xhr.send(JSON.stringify({
27          url: url,
28          un: username
29      }));
30  });
```

APPENDIX B

SAMPLE JSON SNIPPET FOR A POST WITH SHORTCODE --_dmxx4lw

This Appendix B provides an example JSON snippet for the post with the shortcode --_dMXx4Lw. The location within the JSON that contains the post owner's username can be found on line 14 inside the owner object. This is discussed in Section 4.3.2.


```
{
  "qs": [...]
  "static_root": [...]
  "entry_data": {
    "PostPage": [
      {
        "__query_string": "?",
        "media": {
          "code": "--_dMXx4Lw",
          "dimensions": {...},
          "shared_by_author": false,
          "usertags": {...},
          "owner": {
            "username": "blakelively",
            "full_name": "Blake Lively",
            "requested_by_viewer": false,
            "followed_by_viewer": false,
            "has_blocked_viewer": false,
            "profile_pic_url": "https://web.archive.org/web/20151207180858/https://scontent.cdninstagram.com/hphotos-xaf1/t51.2885-19/11078986_655714111224411_1488883663_a.jpg",
            "is_unpublished": false,
            "blocked_by_viewer": false,
            "id": "1437529575",
            "is_private": false
          },
          "comments": {...},
```

```
            "is_ad": false,
            "caption": [...]
            "likes": {...},
            "date": 1449477651,
            "is_video": false,
            "id": "1134623239222821616",
            "display_src": [...]
            "location": null
          },
          "__get_params": {},
          "__path": "/p/--_dMXx4Lw/"
        }
      ]
    },
    "hostname": "www.instagram.com",
    "platform": "web",
    "qe": {...},
    "display_properties_server_guess": {...},
    "country_code": "US",
    "language_code": "en",
    "gatekeepers": {...},
    "config": {...},
    "environment_switcher_visible_server_guess": true
}
```

APPENDIX C

PYTHON CODE FOR add_un_idx API ENDPOINT

This Python code defines the app instance for add_un_idx API endpoint and functions for extracting the username. Lines 2-18 shows the code for the add_un_idx endpoint of the Flask API. The two functions add_to_index_wsd (lines 21-43) and add_to_index_wsd (lines 45-51) serves the purpose of parsing the JSON to extract the username. Once the username is extracted, it will be added to the index. This code is used in Section 4.3.2.

```
1  @app.route('/add_un_idx', methods=['POST'])
2  def add_un_idx():
3      input_json = request.get_json(force=True)
4      raw_urim = input_json["url"]
5      shortcode = raw_urim.rsplit("/",2)[1]
6      username = input_json["un"]
7      success,recordExist = add_to_index_wsd(shortcode,username)
8      if success:
9          response = make_response("<h1>Success, Username and Shortcode
           ↪ Added to the Database</h1>")
10         response.status_code = 200
11     else:
12         if recordExist:
13             response = make_response("<h1>Username and Shortcode
               ↪ Already Exist in the Database</h1>")
14             response.status_code = 200
15         else:
16             response = make_response("<h1>Oops, Looks Like Something
               ↪ Went Wrong</h1>")
17             response.status_code = 500
18     return response
19
20 # Parsing window_shared_data
21 def add_to_index_wsd(shortcode,res):
22     try:
```

```
23          username = get_username_wsd(res)
24          con,cur = connect_DB()
25          cmd = f"SELECT * FROM IG_Index WHERE username='{username}'AND
             ↪  shortcode='{shortcode}'"
26          res = cur.execute(cmd)
27          result = res.fetchone()
28          if result:
29              #print("Record already exist")
30              success = False
31              recordExist = True
32
33          else:
34              # print("Record does not exist")
35              cmd = f"INSERT INTO
                 ↪  IG_Index(username,shortcode,added_date) VALUES
                 ↪  ('{username}','{shortcode}', CURRENT_TIMESTAMP)"
36              res1 = cur.execute(cmd)
37              con.commit()
38              success = True
39              recordExist = False
40      except:
41          success = False
42          recordExist = False
43      return success,recordExist
44
45  def get_username_wsd(res):
46      a = res.split(' = ', 1)[1].strip(";")
47      b = a.split('owner',1)[1]
48      c = b.split('}',1)[0].strip('":') + "}"
49      json_data = json.loads(c)
50      username = json_data['username']
51      return username
```

APPENDIX D

EXTRACTING USERNAME FROM LIVE WEB INSTAGRAM POST

We used the code below to extract the username from live web Instagram post pages. Lines 26-27 illustrate `{"name":"twitter:title"}` and `{"property":"og:description"}`, the two `meta` fields that can be used to extract the username. The usage of this code is explained in Section 4.3.3.

```
1  import glob
2  from bs4 import BeautifulSoup
3  import re
4  import os
5
6  path = "HTML_out/*.html"
7
8  if __name__ == "__main__":
9    with open("extracted_data_2.tsv","w") as f:
10     for file in glob.glob(path):
11       shortcode = file.split("/")[-1].split(".")[0]
12       print(shortcode)
13       size = os.path.getsize (file)
14       with open(file) as g:
15         if size == 0:
16           tusername = username = meta_con = title_con = image_con =
              ↪  des_con = twitter_con = "x"
17           username2 = "size_zero"
18         else:
19           html = g.read()
20           tusername = "x"
21           twitter_con = "x"
22           soup = BeautifulSoup(html, 'html.parser')
23           meta = soup.find("meta", attrs={'name': 'description'})
24           title = soup.find("meta",  {"property":"og:title"})
25           image = soup.find("meta",  {"property":"og:image"})
```

```
26            description = soup.find("meta",
               ↪  {"property":"og:description"})
27            twitter = soup.find("meta",  {"name":"twitter:title"})
28            if twitter:
29              twitter_con = twitter["content"]
30              pattern = "\((@[^\)]+)\)"
31              m = re.search(pattern,twitter_con)
32              if m:
33                match = m.group()
34                tusername = match.strip("()@")
35              else:
36                pattern2 = "(@[^\s]+)"
37                m2 = re.search(pattern2,twitter_con)
38                if m2:
39                  try:
40                    tusername = m2.group().strip("@")
41                  except:
42                    tusername = "x"
43                    username2 = "GROUP_ERR"
44                else:
45                  username = "x"
46                  username2 = "NoMatch"
47            if description:
48              meta_con = meta["content"]
49              title_con = title["content"]
50              image_con = image["content"]
51              des_con = description["content"]
52              #pattern1: (@username) in the og:description tag Ex: David
                 ↪  Beckham (@davidbeckham) on Instagram: "Good morning
                 ↪  and hello!.....
53              pattern1 = "\((@[^\)]+)\)"
54              m = re.search(pattern1,des_con)
55              if m:
56                match1 = m.group()
57                username = match1.strip("()@")
58                username2 = "Pattern1"
59              else:
```

```
60                pattern2 = "(@[^\s]+)"
61                m2 = re.search(pattern2,des_con)
62                if m2:
63                  try:
64                    #print(m2)
65                    match2 = m2.group().strip("@")
66                    username = re.sub("'s","",match2)
67                    username2 = "Pattern2"
68                  except:
69                    username = "x"
70                    username2 = "GROUP_ERR"
71                else:
72                  username = "x"
73                  username2 = "NoMatch"
74              elif meta:
75                meta_con = "HasMeta"
76                title_con = "HasMeta"
77                image_con = "HasMeta"
78                des_con = "HasMeta"
79                username = "x"
80                username2 = "HasMeta"
81              else:
82                meta_con = "NoDes_Or_Meta"
83                title_con = "NoDes_Or_Meta"
84                image_con = "NoDes_Or_Meta"
85                des_con = "NoDes_Or_Meta"
86                username = "x"
87                username2 = "NoDes_Or_Meta"
88              #if no twitter:title tag, use the og:description tag
                 ↪ username if available
89              if tusername == "x":
90                tusername = username
91            meta_con = '"' + meta_con.replace("\n"," ") + '"'
92            title_con = '"' + title_con.replace("\n"," ") + '"'
93            image_con = '"' + image_con.replace("\n"," ") + '"'
94            des_con = '"' + des_con.replace("\n"," ") + '"'
95            twitter_con = '"' + twitter_con.replace("\n"," ") + '"'
```

```
96          line = f"{shortcode}\t{tusername}\t{username}\t{username2}\t{siz⌋
             ↪  e}\t{meta_con}\t{title_con}\t{image_con}\t{des_con}\t{twitte⌋
             ↪  r_con}\n"
97          f.write(line)
```

APPENDIX E

EXTRACTING USERNAME FROM LIVE WEB INSTAGRAM ACCOUNT

Highlighted portions in the HAR data are the extracted fields using the Haralyzer[1] that are required to compile the dataset of individual posts. The dataset and the full HAR file is available in GitHub.[2] The usage of this data is explained in Section 4.3.3.


```
"shortcode": "CEav2jYB12H",
      "edge_media_to_comment": {
         "count": 7596,
         "page_info": {
            "has_next_page": true,
            "end_cursor": ""
         }
      },
      "edge_media_to_sponsor_user": {
         "edges": []
      },
      "comments_disabled": false,
      "taken_at_timestamp": 1598585366,
      "edge_media_preview_like": {
         "count": 1447080,
         "edges": [{
            "node": {
               "id": "2016325463",
               "profile_pic_url": "https://instagram.forf1-3.fna.fbcdn.
                  net/v/",
               "username": "hiru_savi"
            }
         }]
      },
```


[1] https://haralyzer.readthedocs.io/

[2] https://github.com/himarshaj/MSThesis_ArchivedIGPosts/tree/main/KatyPerryLive

```
24        "owner": {
25            "id": "407964088",
26            "username": "katyperry"
27        },
28        "location": {
29            "id": "111670616996681",
30            "has_public_page": true,
31            "name": "Promo Smiles",
32            "slug": "promo-smiles"
33        }
```

APPENDIX F

JSON SNIPPET - ARCHIVED ACCOUNT PAGES (URI-MS)

JSON snippet showing where the post data is available. Lines 30-51 show the data (such as the shortcode, date, comments, count, caption, and likes) for the first post, and lines 62-83 show similar data related to the 12th post. This usage of JSON snippet to extract the posts from an archived account page is explained in Section 4.3.3.


```
{
  "country_code": "US",
  "language_code": "nl",
  "gatekeepers": {...},
  "qs": [...]
  "show_app_install": true,
  "static_root": [...]
  "platform": "web",
  "hostname": "www.instagram.com",
  "entry_data": {
    "ProfilePage": [
      {
        "user": {
          "username": "blakelively",
          "follows": {...},
          "requested_by_viewer": false,
          "followed_by": {...},
          "country_block": null,
          "has_requested_viewer": false,
          "external_url_linkshimmed": [...]
          "follows_viewer": false,
          "profile_pic_url": [...]
          "id": "1437529575",
          "biography": [...]
          "full_name": "Blake Lively",
          "media": {
```

```
            "count": 270,
            "page_info": {...},
            "nodes": [
              {
                "code": "BGltI0Dx4EP",
                "date": 1465809056,
                "dimensions": {
                  "width": 750,
                  "height": 750
                },
                "comments": {
                  "count": 413
                },
                "caption": [...],
                "likes": {
                  "count": 196537
                },
                "owner": {
                  "id": "1437529575"
                },
                "thumbnail_src": [...]
                "is_video": false,
                "id": "1271620987548827919",
                "display_src": [...]
              },
              {"code": "BGlshRlR4C9", [...]},
              {"code": "BGjRubgR4AD", [...]},
              {"code": "BGhdLsuR4Kb", [...]},
              {"code": "BGewBN3x4LU", [...]},
              {"code": "BGciD0xx4CP", [...]},
              {"code": "BGch516x4Bt", [...]},
              {"code": "BGXcvPBx4NQ", [...]},
              {"code": "BGURoZER4J2", [...]},
              {"code": "BGNJAP4R4HK", [...]},
              {"code": "BGHmttyx4FU", [...]},
              {
                "code": "BGFRrDwR4OE",
```

```
                "date": 1464720914,
                "dimensions": {
                  "width": 640,
                  "height": 640
                },
                "comments": {
                  "count": 249
                },
                "caption": "@sagevaughn ...happy sigh ïf¡ïf¡",
                "likes": {
                  "count": 208496
                },
                "owner": {
                  "id": "1437529575"
                },
                "thumbnail_src": [...]
                "is_video": false,
                "id": "1262492996306699140",
                "display_src": [...]
              }
            ]
          },
          "blocked_by_viewer": false,
          "followed_by_viewer": false,
          "is_verified": true,
          "has_blocked_viewer": false,
          "is_private": false,
          "external_url": [...]
        }
      }
    ]
  },
  "qe": {...},
  "display_properties_server_guess": {...},
  "config": {...},
  "environment_switcher_visible_server_guess": true}
```

APPENDIX G

EXTRACT USERNAME FROM ARCHIVED ACCOUNT PAGES (URI-MS)

This Python code is used to extract the post data from account URI-Ms. Out of 3854, we were able to extract posts from 2655 URI-Ms.

```
from bs4 import BeautifulSoup
from urllib.parse import urlparse
import requests
import json
import re

#Recent URI-Ms
def get_shortcodes_case1(json_res,out_file,urim):
  # urim_data = {}
  # range_max = json_res['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['count']
  # print(int(range_max))
  for j in range(0,12):
    # id = str(j)
    shortcode = json_res['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges'][j]['node']['shortcode']
    display_url = json_res['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges'][j]['node']['display_url']
    taken_at_timestamp = json_res['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges'][j]['node']['taken_at_timestamp']
    is_video = json_res['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges'][j]['node']['is_video']
    # shortcode_list.append(shortcode)
    out_file.write(f"{urim},{shortcode},{display_url},{taken_at_timestamp},{is_video}\n")
```

```
20      out_file.flush()
21
22  #2016 URI-Ms
23  def get_shortcodes_case2(json_res,out_file,urim):
24    for j in range(0,12):
25      # id = str(j)
26      shortcode = json_res['entry_data']['ProfilePage'][0]['user']['medi
        ↪  a']['nodes'][j]['code']
27      display_url = json_res['entry_data']['ProfilePage'][0]['user']['me
        ↪  dia']['nodes'][j][]'display_src']
28      taken_at_timestamp = json_res['entry_data']['ProfilePage'][0]['use
        ↪  r']['media']['nodes'][j]['date']
29      is_video = json_res['entry_data']['ProfilePage'][0]['user']['media
        ↪  ']['nodes'][j]['is_video']
30      # shortcode_list.append(shortcode)
31      out_file.write(f"{urim},{shortcode},{display_url},{taken_at_timest
        ↪  amp},{is_video}\n")
32      out_file.flush()
33
34  if __name__ == "__main__":
35    with open('urim_list.txt', 'r') as f:
36      urim_list = f.readlines()
37    with open('extracted_post_info.csv', 'w') as g:
38      # g.write("category,url,hostname,discription\n")
39      g.write("urim,shortcode,display_url,taken_at_timestamp,is_video\n")
40      for URIM in urim_list:
41        try:
42          URIM = URIM.strip("\n")
43          datetime = re.findall('\d{14}',URIM)
44          # print(datetime)
45          response = requests.get(URIM)
46          html = response.text
47
48          # with open('URIM.html', 'r') as h:
49          #   html = h.read()
50
51          filename= "output_urim_html/" + datetime[0] + ".html"
```

```
52          with open(filename, 'w') as h:
53            h.write(html)
54
55          #print(html)
56          soup = BeautifulSoup(html, 'lxml')
57          body = soup.find('body')
58          script_tag = body.find('script', text=lambda t:
             ↪  t.startswith('window._sharedData'))
59          raw_string = script_tag.string.split(' = ', 1)[1].rstrip(';')
60          json_string = json.loads(raw_string)
61          # print(json_string)
62          get_shortcodes(json_string,g,URIM)
63          print(URIM,"++DONE++")
64        except Exception as e:
65          print(URIM,"++ERROR++",e)
```

APPENDIX H

EXTRACTING FROM WARC

This Python code is used to collect the {username,shortcode} pairs from WARC files in our test sample to be added to the index. We used two regular expressions (lines 14 and 22 to 27) to match and extract the username from the payload. This method is discussed in Section 4.3.3.

```
1  #!/usr/bin/env python3
2
3  from warcio.archiveiterator import ArchiveIterator
4  import sys
5  import json
6  import re
7  from datetime import date
8  from datetime import datetime
9
10 file = sys.argv[1]
11
12
13 def case_1(payload,shortcode):
14     pattern = '"owner":{"username":"[a-zA-Z0-9._]+"'
15     m = re.search(pattern, payload)
16     match = m.group(0)
17     username = match.split(":")[2].strip('"')
18     return username
19
20
21 def case_2(payload,shortcode):
22     pattern = '{"graphql":{"shortcode_media":{"__typename":"GraphImage
       ↪  ","id":"\d","shortcode":"%s"' % shortcode
23     m = re.search(pattern, payload)
24     match = m.group(0)
25     a,id_str,code = match.split(",")
26     id = id_str.split(":")[1].strip('"')
```

```
27      pattern2 = 'owner":{' + id_str + ',"is_verified":true,"profile_pic
        ↪  _url":".*","username":"[a-zA-Z0-9._]+"'
28      m2 = re.search(pattern2, payload)
29      match2 = m2.group(0)
30      id_n, verified, propic, username = match2.split(",")
31      username = username.split(":")[1].strip('"')
32      return username
33
34  if __name__ == "__main__":
35      with open(file, 'rb') as fh:
36          for record in ArchiveIterator(fh):
37              payload = record.content_stream().read()
38              payload = payload.decode("utf-8")
39              postURI = record.rec_headers.get_header('WARC-Target-URI')
40              date = record.rec_headers.get_header('WARC-Date')
41              shortcode = postURI.split("/p/")[1].strip("/")
42              try:
43                  extracted_username = case_1(payload, shortcode)
44              except Exception as e:
45                  extracted_username = case_2(payload,shortcode)
46              print(shortcode, extracted_username)
```

APPENDIX I

PYTHON CODE TO OBTAIN CORRECT CASE SHORTCODES

Using the response from CDX API with `prefix` match we obtained the shortcode with the correct case. This Python code is used to obtain the correct case for the shortcode (URI-R prefix "https://www.instagram.com/p/d/" of shortcode `d` will return the URI-Ms with the correct cased shortcode `D`.

```
#!/usr/bin/env python3

import io
import sys
import csv
import os

url_prefix = "https://www.instagram.com/p/"

if __name__ == "__main__":
  with open("final_shortcodes.txt", "r") as f:
    shortcodes = f.readlines()

  with open("correct_shortcode.csv", "w") as f:
    for shortcode in shortcodes:
      shortcode = shortcode.strip("\n")
      shortcode = shortcode.strip('"')
      ig_url = url_prefix + shortcode  + "/"
      ig_url = ig_url.strip("\n")
      url =  "http://web.archive.org/cdx/search/cdx?url=%s&matchType=p
       ↪  refix&limit=1" % ig_url
      prefix = "https://web.archive.org/web/"
      awk = '{print "https://web.archive.org/web/" $2 "/" $3};'
      cmd = "curl -s \"%s\" | awk '%s'" % (url,awk) #use this for
       ↪  prefix search
      out = os.popen(cmd)
```

```
25          urim = out.read()
26          if urim == "":
27            correct_shortcode = None
28          else:
29            correct_shortcode = urim.split("/")[9]
30            correct_shortcode = correct_shortcode.strip("\n")
31          f.write(f"{shortcode},{correct_shortcode}\n")
32          f.flush()
```

APPENDIX J

FLASK API ENDPOINTS - app.py

This Python code is for defining the app instance for the two API endpoints that support reading from the IG_Index. The `get_posts` endpoint allows the user to get all the posts that belong to a particular Instagram account and the `get_username` endpoint allow retrieving the username of a particular shortcode if it is available in the index. The functions that defined the `add_un_id` API endpoint to automatically add the extracted shortcode and username to the IG_Index are shown in Appendix A. This is discussed in Section 4.3.1.

```
1  import os
2  import logging
3  from flask import Flask, render_template, request, url_for, jsonify,
    ↪  make_response
4  import json
5  from flask_cors import CORS
6  import sqlite3
7
8  logging.basicConfig(level=logging.INFO)
9
10 app = Flask(__name__)
11 CORS(app)
12
13 def read_index_get_posts(user):
14     con, cur = connect_DB()
15     cmd = f"SELECT shortcode, added_date FROM IG_Index WHERE
        ↪  username='{user}'"
16     res = cur.execute(cmd)
17     shortcodes = res.fetchall()
18     if shortcodes:
19         shortcodes_list = []
20         for each in shortcodes:
21             shortcodes_list.append(each[0])
22         sc = ','.join(shortcodes_list)
```

```
23          success = True
24      else:
25          sc = "null"
26          success = False
27      return success, sc
28
29  def read_index_get_username(sc):
30      con, cur = connect_DB()
31      cmd = f"SELECT username FROM IG_Index WHERE shortcode='{sc}'"
32      res = cur.execute(cmd)
33      user = res.fetchone()
34      if user:
35          success = True
36          username = user[0]
37      else:
38          success = False
39          username = "null"
40      return success, username
41
42  @app.route('/get_posts', methods=['GET'])
43  def get_posts():
44      user = request.args.get("username")
45      success,shortcodes = read_index_get_posts(user)
46      return jsonify(username=user , shortcodes=shortcodes)
47
48  @app.route('/get_username', methods=['GET'])
49  def get_username():
50      sc = request.args.get("shortcode")
51      success,user = read_index_get_username(sc)
52      return jsonify(username=user , shortcode=sc)
53
54  if __name__ == '__main__':
55      app.run(debug=True)
```

VITA

Himarsha R. Jayanetti
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION

2020-Present, Ph.D. in Computer Science, Old Dominion University, Norfolk, VA.
2019-2023, M.S. in Computer Science, Old Dominion University, Norfolk, VA.
2013-2017, B.Eng in Computer Science, Gujarat Technological University, India.

PROFESSIONAL EXPERIENCE

2019-Present, Graduate Research / Teaching Assistant, Old Dominion University.
2017-2019, Network Support Engineer, Exetel Telecommunications, Colombo, Sri Lanka.
2016, Research Intern, Department of National Archives, Sri Lanka.

CONFERENCE PRESENTATIONS

1. "Robots Still Outnumber Humans in Web Archives, But Less Than Before" presented at the Theory and Practice of Digital Libraries (TPDL) 2022 in Padua, Italy.
2. "Creating Structure in Web Archives With Collections: Different Concepts From Web Archivists" presented at the Theory and Practice of Digital Libraries (TPDL) 2022 in Padua, Italy

PUBLICATIONS

1. A complete list is available at `https://himarshaj.github.io/publications/`