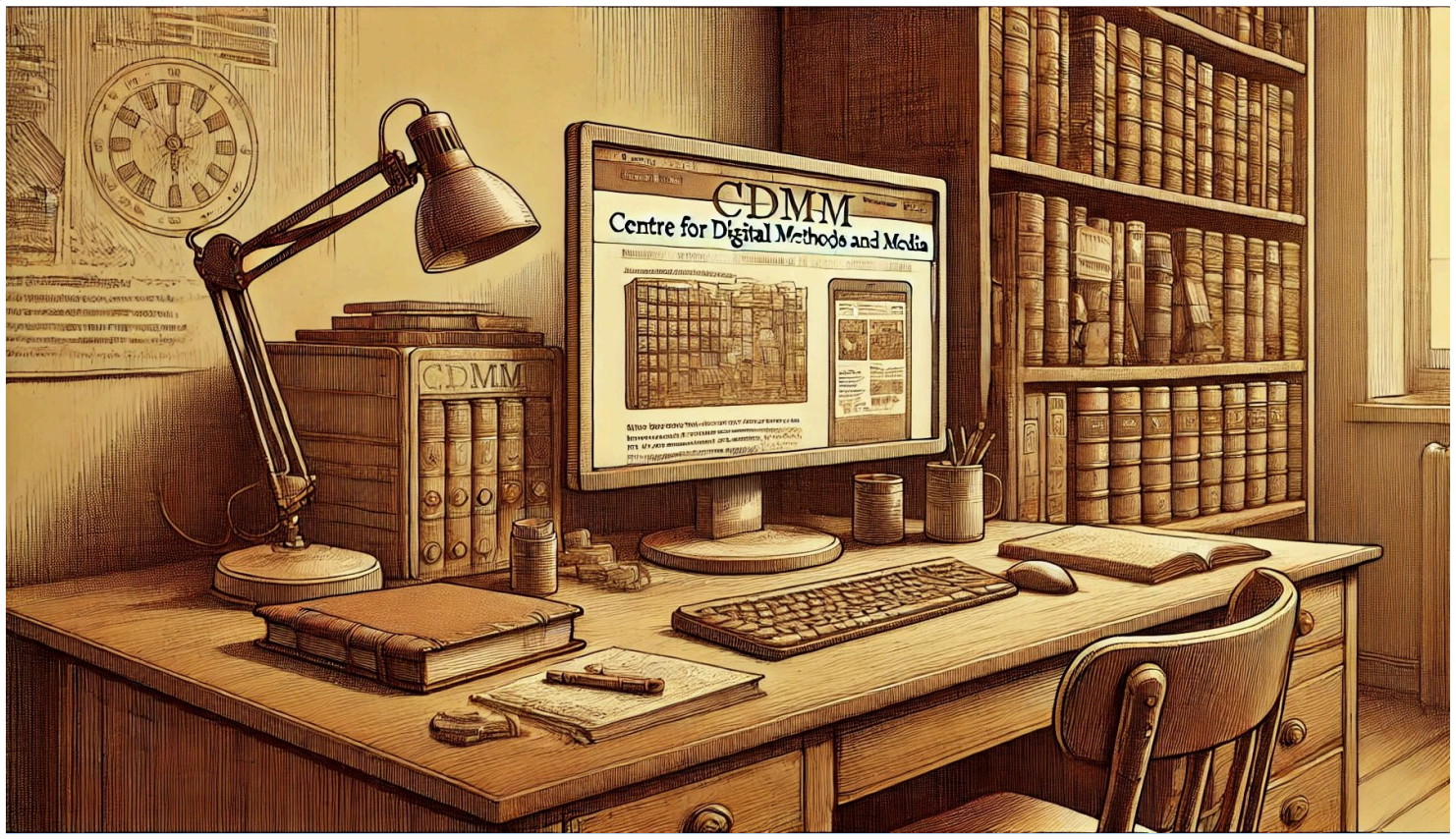
Asger Harlung

Web Archives and Web Archiving

Introduction for Scholars and Students



CDMM Publications

Web Archives and Web Archiving

Introduction for Scholars and Students

Asger Harlung

Web Archives and Web Archiving
Introduction for Scholars and Students

CDMM Publications

2025

Asger Harlung:
Web Archives and Web Archiving
– Introduction for Scholars and Students

Illustrations and screenshots by Asger Harlung, except figures 6, 12, 13, 38, by Niels Brügger, and figures 7, 8 (with a post edit by the author), 14, 15, 16, 17, 18, by Janne Nielsen.

Cover design by Asger Harlung.

Cover illustration, “Bridging the Old and the New” by Asger Harlung with the aid of ChatGPT3 in six iterations (2024).

PDF edition: ISBN 978-87-975247-0-1

Published by
CDMM Publications
Helsingforsgade 14
8200 Aarhus N.

https://cc.au.dk/en/cdmm

Thanks

The author wishes to extend specials thanks to four persons:

Niels Brügger. Professor in Media Studies, and my leader from 2016 to 2024. The nicest leader that anyone could wish for, and a leading international voice in the field of web archiving in research for over twenty years who I am happy and proud to have worked with.

Janne Nielsen. Associate Professor. A close colleague from 2016-2022, and author of the course book mentioned in the preface, which helped me learn many of the first important insights on archived web. It was good times when we taught together at researcher workshops.

Anders Klindt Myrvoll. Programme Manager at Netarkivet. For collaboration over the years, and for taking the time for demonstrations and helping to clarify details concerning Netarkivet.

Gitte Harlung. My wife. For the last 28 years she has read my work providing proofreading with fresh eyes, as well as valuable feedback on legibility. If this book were of a more personal nature, it would have been dedicated it to her.

Contents

Preface

This book was originally intended to be a co-authored update of “Using Web Archives in Research – an Introduction”, by Janne Nielsen (2016); a course book published by the national Danish infrastructure for archived web and internet studies “NetLab” (2012-2022).

For a number of practical reasons that plan was dropped, and this book takes on the task of introducing web archives and web archives for research purposes in a fully new volume.

But the author of the book you are now reading owes a great debt to Janne Nielsen, a close colleague in the years we worked together in “NetLab”. Her book was an important first introduction to the field, and we frequently taught together at workshops for researchers.

The reasons for a new introduction are ongoing developments in the field, and the intention of covering an additional number of challenges and practicalities; e.g. advice for finding software and services, attempts to address how the field moves, a chapter on the big challenges with social media, and changes in large, institutional archives towards a more research-oriented inclusion of metadata and other initiatives for supporting research, rather than merely preserving content.

The book is aimed primarily at researchers and students in the humanities and social sciences, who want to include online content or trends in contemporary research. It may of course also serve others with similar interests and purposes.

The approach is “low tech” in the sense that the focus is on helping the reader to understand the online and archived types of content, making sense and use of them, and to get an idea of how to create local collections for specific topics and purposes without going into programming and other high level areas of computer expertise.

The book cannot fully cover a perpetually moving field, and it cannot omit being “technical” since all research will by necessity need a basic understanding of the nature and inner workings of the media it will look into.

A lot of advice and information that proved useful or relevant during researcher workshops and PhD Courses in 2016-2022 has been included, following a philosophy of better providing too much information than too little. The book will therefore contain many details that may be of interest to one reader, but unimportant to another.

In an attempt to avoid becoming unnecessarily frustrating on the technical part, a section of “takeaways” is presented at the start of each chapter. They may hopefully serve as a way to navigate and prioritise the reading process to better meet the reader’s personal and present interests.

The emphasis is on the purpose of providing a practical introduction to web archiving and web archives. As such, this book is not a research work and will not rely heavily on referring to scientific sources except for a few citations. Basic and easily verified facts will not be supported with references.

Since web services and technologies are extremely transient, subject to constant and ongoing change, this book aims at addressing principles, methods, advantages and challenges on a general level, rather than providing specific advice for specific technologies.

Any examples given, especially in the form of illustrations, are thus meant only as examples in order to make a point clearer, while the exact observations, software, or methods that may be mentioned are fixed in time and very much subject to change.

This book is not perfect. It cannot be. But I do hope it may help.

Asger Harlung, 2025

How to Use This Book and Target Audience

This book is primarily meant as a guide or introduction to researchers and students in the humanities and the social sciences. It may be relevant to others; in other fields of study or lines of work, such as librarians, journalists, authors, or anyone to whom historical use of web data would be of interest. All readers are welcome.

The book is meant as an introduction to the field of web archives and web archiving, with emphasis placed on practical use and basic understanding.

As such, it may be read from cover to cover in order to get a broad picture. Readers who may choose to do so are kindly asked to be forgiving: For some readers the large amounts of details may make for a somewhat strenuous experience, but for others perhaps and hopefully a satisfyingly broad introduction.

Another way to read this book could be called chapter-jumping, somewhat similar to surfing on the Web. The “Takeaways” pages at the start of the main chapters give the highlights for that chapter, by which it should be possible to dive into the chapters that are of the highest immediate interest. The book is heavily cross-referenced, so that details that are explained in depth elsewhere may be followed up upon.

All readers are recommended to read the takeaways pages for all chapters in order to get an overall impression of the field that this book will address, and determine their priorities for the reading process.

For anyone interested in making use of archived web, it is also recommended to experiment with actively doing some archiving oneself. This will provide insights and a better understanding of the nature of archived web and the impact of practical issues that cause archived web to differ in some ways from the originals it was taken from.

1 The Importance of Web Archiving

Takeaways

♦ More and more data is produced in a digital form, in amounts that vastly surpass physical and analogous types of data.

♦ Data published on the World Wide Web (the Web) changes or disappears rapidly, and remains available only if it is preserved (archived) before it disappears.

♦ Web archives with copies of Web data and references are necessary resources for most types and topics of humanities research that will look into trends or phenomena after the start of the digital age.

The World Wide Web was invented in 1989 by Tim Berners-Lee at CERN, and was opened to the public in 1991. It is the phenomenon that people today know in the form of websites and web pages. It will henceforth be referred to as “the Web”.

In daily language the Web is often referred to as “the web”, or “the Internet”, which it is part of, but not identical with. “The Internet” is a global system of computers linked in a large network sharing information of very different types, using a large number of “protocols”; that is, standards for various types of information exchange.

As a few examples of protocols that will likely be recognisable to the reader, and that are *not* “the Web” are email protocols (POP3, SMTP, IMAP), file transfer protocols (FTP), streaming protocols (RMTP, WebRTC, SRT, FTL, etc.).

The protocol invented for and defining the Web is the HTTP transfer protocol. The protocol makes it possible to create websites and web pages that may be accessed and read by a wide public, and it is this form of information sharing that defines the Web. Information sharing is here understood in the sense of (digital) data, without regard for whether the purpose of the data in a specific case is true, artistic, deliberately false, incomplete, misleading, misunderstood, etc.

Since the emergence of the Web and its fast development to a widely known and used system for information sharing and gathering it has been a commonplace misunderstanding that the Web is a sort of archive in itself. After all, one can easily get the impression that it is possible to find information on “everything” and that such information “is always present”.

But regarding the Web as an archive would be a mistake, because the content of the Web is very, very far from stable, as it would be in an archive. As easy as it is to find information, it is just as easy to lose that specific version of the information again. A simple example that should be recognisable to most people is having stored a Web

address, e.g. in an article or in a bookmark – and one day finding that the web page that this address points to no longer exists.

Research has made it clear that the Web is indeed an extremely transient medium, where information constantly changes, and comes and goes.

The problem may be illustrated with these examples of research results:

♦ The survival survey revealed that more than 90% of the web pages had disappeared in the last 12 years. The life span study found that the average life span of a web page is 1,132.1 days (Agata et al. 2014, p. 464).

♦ 50% of resources [are] unrecognisable or gone after 1 year, 60% after 2 years, 65% after 3 years (Jackson 2015, p. 20).

♦ In 2023 a large-scale study was conducted by Kritika Garg, Sawood Alam, Michele Weigle, Michael Nelson, and Dietrich Ayala. 27.3 millions URLs archived from 1996-2021 were compared to a crawl in 2023. The study found that URLs have an average lifespan of 2.3 years. Root URLs (main website addresses) have an average lifespan of 8.8 years, and the average lifespan for deep URLs (subpages on websites) is 1.3 years. (Weigle 2024)

The implications of such observations are important, and severe:

If one wants to study online content over a number of years, the original web addresses will more likely than not be useless in most cases. (Web addresses should more correctly be referred to as URLs; this is discussed in the chapter 4.1.2 URLs: Uniform Resource Locators).

The same goes for references in the form of giving the web address with a retrieval date; e.g. “https://cc.au.dk/en/cdmm/tools-and-tutorials/legal-framework/social-media-terms-of-service (retrieved at

2024.06.11)". Such a reference will in many cases become useless as verification for the claims that it was given to support. Possible solutions for better referencing will be discussed in the subchapter 8.3 Referencing from Web Archives.

Web archives and web archiving are therefore the preservation of content which is otherwise likely to change or disappear.

In order to appreciate the scope and implications of this, it is worth taking a look at digital communication and media as data resources compared to the printed and analogous materials that by long-standing tradition have been the primary focus for many types of studies in the disciplines of the humanities and the arts.

The following observations from a time span of just twelve years may serve:

- 2000: 75% of the world's data was stored in analog form (paper, film, photographic prints, vinyl, magnetic cassette tapes, etc.).
- 2007: 7% analog, 93% digital.
- 2012: Only 2% of all stored data was stored in analog form (Mayer-Schönberger & Cukier 2013, p. 8-9).

The trend of digital data overtaking the leading position of the data produced and stored globally must be seen in perspective:

It does not imply the decrease or "the end" of e.g. printed content; rather it shows that an explosive development has taken place in the production of digital data. It is not that physical or analogous data are disappearing or becoming a thing of the past, but rather an observation that for all non-digital content produced, increasingly larger amounts of digital content is being produced at the same time.

Depending on interests or tastes, the amounts of digital data are also of very relative relevance. Large amounts of the digital data being

produced are produced by non-professionals and non-experts. For example, huge amounts of videos being posted on YouTube on a daily basis count as data[1], but the importance and value one may place with such data depends on topics, interests, and degrees of seriousness, quality and qualifications.

Imagine a short video of two minutes explaining the basic rules of addition and measuring 20 megabytes (Mb), versus a digital educational book on mathematics measuring 5 Mb. The information value in respect of understanding mathematics is surely higher in the book than in the video, but the information value of the video is four times higher than the book if measured (primarily) as image and audio data.

Data is data, but the informational value of any amount of data depends entirely on the criteria that it is measured against.

Nevertheless, there is no way around the fact that in respect of pure amounts digital data now represents the majority of data produced daily, weekly, or annually.

This means that almost anyone who wants to study modern trends, modern communication, modern interactions between people, etc. in almost any perspective, will have to look into trends and interactions that were both created and published in a digital form.

Therein lies the problem: Books, newspapers, paintings, photographic film and prints, and other forms of analog content do not quickly cease to exist, but most of the digital content found on the Web does. That is, unless it is preserved in a web archive.

What this implies is that modern scholars who want to study contemporary topics simply have to include content that was created digitally and distributed online on the Web – and that in order to do so

[1] As of June 2022, more than 500 hours of video were uploaded to YouTube every minute (Ceci 2024).

they will need web archives where the relevant content has been preserved.

The transient, constantly changing nature of the Web points towards at least five good reasons for archiving web content:

- To preserve cultural heritage.
- To preserve web materials as stable research objects.
- To be able to document and illustrate claims based on analyses of web materials, whether the web content is in itself the research object or a source of knowledge about other research objects.
- To provide modern and stable source references.
- To serve as documentation in general or legal claims.[2]

[2] Copies of web pages from web archives are at an increasing degree becoming accepted as court evidence, although practice has varied. See for example Pembroke-Birss (2021) in the reference list.

2 Definition of Web Archiving

Takeaways

♦ Web archives are repositories of content which has been preserved from the live web, as opposed to archives of any other type of content that may be offering online services with digital copies.

♦ Web archives differ from other types of archives by not preserving originals, but rather copies and reconstructions of originals.

♦ There are two distinct types of web archives; large-scale "institutional archives" which aim at preserving large amounts on content, e.g. the entire web, or the web domain(s) of a nation – versus smaller-scale and localized archives created by one or more researchers in order to preserve content for specific topics and research purposes.

In this book web archiving is defined as:

”Any form of deliberate and purposive collection and preservation of web material (Brügger 2018, p. 79).”

This definition embraces two types of web archives:

1) Large-scale initiatives such as The Internet Archive which aims at preserving the entire World Wide Web as fully and completely as possible, national web archives, such as the Danish Netarkivet, the Portuguese web archive, Arquivo.pt, and finally large-scale archives for dedicated topics, such as the End of Term Web Archive which specifically preserves federal government websites in the USA.

This type of large-scale initiatives will be referred to as *institutional web archives*, or *institutional archives* in short.

2) Smaller-scale and localized archives of content preserved for specific topics or areas, as data and documentation for specific research interests. “Smaller scale” must be understood in a very broad sense; it may be anything from a collection of a limited number of web pages such as blogs or articles, to a large amount of data that cannot realistically be treated fully by humans without the aid of computational methods, for example a collection of hundreds of thousands of posts in social media, or of a large collection of full websites. But this type of collection is designated by the specific boundaries of the purpose of collecting the materials, and of being created as a localized and limited archive for a specific purpose. Localised refers to the person or group for whom such an archive is created, rather than a geographical localization.

This type of limited archives will be referred to as *local archives*.

2.1 Digitised, Born Digital, or Reborn Digital

There is a typological definition with implications that are connected specifically to the phenomenon of archived content in online archives, libraries and collections.

It is the distinction between three main types of digital media or content:

Digitised, born digital, and reborn digital.

In the introduction to "*The Archived Web: Doing History in the Digital Age*" Niels Brügger gives the following definition:

"(…) it may be argued that digital media may be grouped under three major headings, depending on their provenance – that is, on how they became digital. These headings include digitized media (nondigital media that have been digitized), born-digital media (media that have not existed in any form other than digital), and reborn digital media (born-digital media that have been collected and preserved, and that have been changed during this process) (Brügger 2018, p:5)".

This distinction between these three types of digital content will be used throughout this book.

The remark that "born-digital media that have been collected and preserved" have also "been changed during this process" is an important observation rather than a criterion in itself:

Any archived copy of a web page, a website or other online content is not an original, but an attempt at preserving the closest possible approximation to the original.

This important understanding will be elaborated upon in 7 The Characteristics of the Archived Web.

2.2 The Term “Web Archive"

Since The Internet Archive was founded in 1996 (see 8.1 The Internet Archive) the term “web archive” has become the established and formal designation for a repository of resources preserved from the World Wide Web.

This may sometimes cause some confusion, especially since other archives such as public records archives, local archives, historical archives, archives for the works and records relevant to artists, etc. have often embraced the web, offering digitised copies of their collections or parts thereof. Such “archives on the web” might logically be assumed to be “web archives”, but by definition they are not.

To persons using the online part of an archive of records that were not born-digital, it may seem logical to assume that a book on web archives might be helpful in finding and handling resources. But a book on web archives such as the one you are presently reading will probably be of little relevance to projects focusing on older and originally analogous types of records, since the term “web archives” specifically covers web content preserved since or after the late 1900s.

There is thus a profound (but not easily guessed) difference between a web archive, which is a collection of content and resources originally published on the web, and any other type of archive which as a part of modern life also has an online presence on the web.

The live and online web may be compared to a large ecosystem with changing and dynamic entities and areas which come and go, change, interact with and depend on one another in varying degrees. The content found here are the dynamic, moving, and changing originals, which by their very nature cannot be taken out of their context and placed in an archive as “originals”.

In the same vein, a web archive may be compared to an exhibition on natural history, where one is likely to see tableaus of taxidermied animals, displayed alone or in reconstructions of their natural habitats, thus getting an impression of natural phenomena that cannot in and of

themselves be displayed, but rather represented, explained and visualised.

The point of this analogy is that whenever someone accesses an archived web page or web site, they are not accessing an original, but something that was copied from a dynamic context, with a loss of that context, and with temporal and technical limitations causing the copy to differ from the original in ways that may be very subtle, or less so.

The implications of this, and the types of temporal and technical limitations in question will be explained in 7 The Characteristics of the Archived Web.

Historically speaking, archives originally mean “a place in which public records or historical materials (such as documents) are preserved” (definition derived from the Merriam-Webster Dictionary)[3] and as such the term “archive” is arguably the most correct one for a preservation of web content.

But also historically speaking, the term “archive” has become associated with physical and analogous originals of documents, letters, blueprints, art, etc., as opposed to libraries where the collections are of materials that have been widely published and distributed.

In that latter sense, web archives differ very strongly from other types of archives. Their content has in common with that of libraries that it has been widely distributed, and more notably that due to the nature of that content they cannot offer the possibility of going back to check the originals – which may be lost, while only copies of varying quality remain.

3 https://web.archive.org/web/20230609073736/https://www.merriam-webster.com/dictionary/archive

3 IT, Scholars in the Humanities, and You?

Takeaways

♦ Gathering, using, or analysing digital data preserved from the web demand a reasonable understanding of the underlying technologies beyond that of everyday computer or internet usage.

♦ All-round IT skills and a basic understanding of the underlying technologies of the online and archived web will go a long way for most projects that will include web archives or web archiving.

♦ Research projects where large amounts of data must be gathered and/or analysed should take the need for IT skills into account, and consider ways of including the necessary level of dedicated expertise in the project framework.

♦ Non-expert IT skills may be defined by ability and readiness to combine different operations and different programs in order to achieve the necessary results.

Websites and web pages offer a spectrum of content types, most of which are recognisable as media types from before the digital age, namely text, images, video, and audio. They also offer various types of animations, effects such as fading content in or out, programs for downloading, online simulations, games, etc.

All of this content relies on layers of programming and technology, by which websites and web pages are a new, different, and digital medium compared to the older and traditional media types. In order to understand what can be preserved and how, and what cannot be preserved and why, and what may be done in the latter case – at least a basic understanding of the underlying technologies is required.

Moreover, if or when web content is archived, further technology is applied in order to do so, and the results are affected by this. Therefore at least a basic understanding of what the archiving technologies do to the original medium, and what changes they thus apply to the preserved content, is also necessary.

In other words, since web content, especially in archived forms, is a necessity for most modern scholars and studies working with the present or the recent past – a certain level of technical insights are, too.

The necessary basic insights can be broken down to three main categories, which will be dealt with in the following main chapters:

- Understanding the Web as a technology (chapter 4 The Web as a Technology).

- Understanding how web archiving takes place and what it does to the content (chapters 5 The Web Archiving Process, 6 Social Media, and 7 The Characteristics of the Archived Web).

- Acquiring the ability to preserve web content oneself (chapter 9 Making Your Own Archive).

The chapters will attempt to present the topics in ways that are accessible without technical interests or dispositions. As an example, chapter 9 Making Your Own Archive will primarily focus on solutions that are easily applicable and do not require programming skills.

The necessity of technical know-how may seem disturbing to some, but the purpose of stating it is obviously not meant to cause such reactions.

One simply cannot study English language texts without acquiring some knowledge of the English language, or mathematical questions without learning the corresponding level of mathematical methodology. Similarly, one cannot study digital content without a level of understanding of its nature. The depth of understanding and the skill sets necessary will vary greatly, depending on what, specifically, needs to be studied, and at which depth.

In her article, “Breaking in to the mainstream: demonstrating the value of internet (and web) histories”, Jane Winters speculates, that:

“The most significant barrier to working with web archives is, quite simply, that it is difficult; it requires skills that many historians do not have, and in the short term may be unwilling to learn; it involves acknowledging a degree of ignorance with which otherwise seasoned researchers may be uncomfortable (Winters 2017, p. 2).”

These thoughts can be taken a bit further by speculating that generally speaking, most scholars in the humanities and social sciences are by inclination and tradition not very interested in IT *per se*.

Projects involving archiving of web content, or making use of or analysing archived web content – may require anything from a basic acquaintance with the underlying technology, to a profound understanding.

It is therefore a very good idea to consider the type(s) of project(s) that one may be undertaking measured against the IT skills that may or will be relevant.

3.1 Considering IT Skills Against Project Complexity

Scholars in the humanities or the social sciences come to fields that they have chosen out of interests that will in most cases not be very IT-heavy, in scientific fields which in themselves have long-standing traditions that certainly are not IT-heavy.

Modern societies and organisations are of course IT-heavy today, using data equipment for communication, documentation, cataloguing, statistics, analyses, production, and other areas of work.

So even scholars specialising in anything that traditionally never required IT skills of any kind, from ancient literature to archeology, from anthropological studies of tribal cultures to trade routes in the Renaissance – are likely to become knowledge-workers and in that role, they too will depend on and work with IT.

That, of course, is not a main interest for such scholars but rather something that may be seen as a mere practicality, a workplace demand of modern life.

After years of working in IT-heavy environments, and working with IT themselves, modern scholars may consider themselves fairly experienced as IT users.

The potential problem here is, that being used to writing documents, sending emails, registering information, and visiting websites do not imply an understanding of what is going on in the computers, or problem-solving skills outside of well-known tasks. But if a scholar wants to do research in contemporary subjects where relevant material is or includes born digital content such as websites or social media, then the mere ability to use a computer for known tasks is unlikely to suffice. Rather, some understanding of what is going on technically, on

websites, on social media, in web archives, and in programs and applications used for data gathering and treatment will be necessary.

This is not to say that scholars working with contemporary topics need to become technical experts such as programmers. But it will be necessary to develop a sound understanding – and with it almost certainly some level of interest – in the technological aspects and handling of the digital material.

It will also be relevant to give serious consideration as to whether the scholar's IT proficiency is sufficient for an envisioned project. If not, then it does not necessarily mean that it must become so, but it stands to reason that the necessary skills for a project must be present for the project to be successfully accomplished. But even if IT experts or skilled students are hired, the scholars themselves must still have an understanding of the nature of the data, its possibilities and its limitations.

Of course, the demands of a project may be modest, without any need for either specialist understanding or help. But there is still a difference between being used to studying websites or web pages, and to treating and understanding them as data. So even a project where the main purpose is to compare the treatment of a given topic on a limited number of websites, will imply a need to understand why and how the websites should be archived for research purposes, and also why and how the archived versions may differ in subtle ways from the originals online.

But modern studies may also be large-scale studies of e.g. the spread of conspiracy theories, or of fake news, or of reactions to large-scale events, or other topics where vast amounts of data are both available and important. The latter type of projects will require a solid understanding of data gathering and treatment, and may require dedicated IT expertise in the more complex and ambitious cases.

All this begs the question of how a researcher may feasibly assess his/her IT skills in a broader sense, and attempt to relate this to a research project.

3.2 A Research-based Background for Discussing IT Skills

IT skills are often talked about or mentioned, but rarely with a sufficiently precise definition or framework for a broader discussion. However, a suitable background is provided in the OECD report “Skills Matter: Further Results from the Survey of Adult Skills” (OECD, 2016).

The following is an explanation of the rationales behind the categories and skill distributions used for discussing the IT skills of scholars and students in the humanities and social sciences in the rest of this chapter.

OECD measures practical IT skills as ”problem-solving [ability] in a technology-rich environment (OECD 2016, p. 21)”.

Users’ IT proficiency levels are thus determined from the number of operations and operators, that the users are able to combine in order to obtain a desired result.

This is a practical approach which reflects what users can actually do, and thus provides a reasonable criterion for understanding the IT skills of users that are not computer specialists.

OECD conducted research from 2013-2015 with 216.250 adults, age 16-65, in 33 OECD countries (OECD 2016, p. 22-22).

The results were categorised in five primary skill levels ranging from “score range not applicable” (divided into three different modes where the test subjects opted out, or e.g. were unable to perform the most basic tasks such as using a mouse, or scrolling down on a web page), over levels “below 1”, “1”, “2”, and “3”,where level 3 is the ability to solve complex tasks that require combining of information, applications

and functions, sorting and handling unforeseen results, etc. (OECD 2016, p. 53).

26.2 % of the test subjects were in the category “score range not applicable”. Furthermore the research showed that the highest age group from 55-65 had a larger percentage of people with low IT skills compared to younger age groups.

42.9 % were basic IT users without IT skills outside of well-known and familiar applications and their use for simple and well-defined tasks. The number combines the aforementioned groups “below level 1” and “level 1” as one major group of persons.

25.7 % were capable of using unfamiliar applications in combination with familiar ones, performing tasks such as sorting or monitoring, and handling unforeseen results.

5.4 % were capable of handling unfamiliar and familiar applications, performing advanced sorting and monitoring tasks, handling high frequencies of unexpected results, and deciding how the unexpected outcomes should be handled, e.g. by assessing and sorting the applied methods and data in order to get a useful outcome.

(Skill distributions above according to OECD 2016, p. 53).

It is fair to assume that as the years pass after the research ended in 2015, the pattern will change, mainly by the number of people with low IT skills decreasing somewhat due to the larger number of better IT skills found in younger age groups.

However, the overall results do not support speculation that IT skills are merely a question of age. In modern societies where IT is present in both daily, private, and work life, the research suggests that a majority of persons have little interest in more complex IT usage, and rather prefer to rely on the most basic functions that they need and are accustomed to in daily life. This would suggest that the overall skill distribution pattern among IT users is likely to remain the same, with

future variations primarily due to a shift in generations, and thus comparatively insignificant. It can be expected that the group of people without IT skills will become somewhat smaller over time, and that the groups of people with IT skills will thus grow. But since this will be a bottom-up process the growth can be expected to be distributed along the patterns already established, therefore not causing a significant change to the distribution of skill levels among IT users.

Modern scholars and students in the humanities and social sciences would not be able to perform all their tasks without any kind of IT use, so in order to better understand this group we can disregard the population group(s) without IT skills.

We can only speculate on whether the group of scholars and students in the humanities and social sciences can be expected to differ from the patterns characterising the rest of the IT users:

On one side, these scholars and students come from parts of the populace that are not technically inclined in their choice of studies or careers, which could imply a statistically lower performance in IT use and skills. On the other side, these scholars and students are knowledge workers, or aiming towards becoming so, which could imply a higher than average distribution of IT skills.

Since the possible variances in IT skills cannot be determined, and could arguably go either way, let us assume that the patterns of IT skills for these scholars and students follow the overall patterns for the populace with the same average distribution as found in the broad populace of IT users.

Keeping in mind a margin for error, the rest of this chapter will discuss IT skills as if this were the case.

3.3 Estimated IT Skill Distribution in Humanities Scholars and Students

Based on the rationales given in the previous subchapter we can now describe a typology with three major skill levels and their distribution among scholars and students in the humanities.

Let us call these “Low” (corresponding to the 42.9 % of basic IT users without a ready capacity for advanced computer use from the previous subchapter), “Average” (corresponding to the 25.7 % of users that can handle data across known and unfamiliar applications), and “High” (corresponding to the 5.4 % of users that can handle data and decide on and perform adjustments across known as well as unfamiliar applications). The terms, “low, average, and high” here refer to the demands on IT usage and task complexity, not to the actual distribution of corresponding skills.

By disregarding people with no IT skills (the 24.3 % mentioned in the previous subchapter) we can get a new distribution number by recalculating the overall distribution to the total distribution of people with IT skills, where the 73.8 % of the populace that can use IT is now set to 100 %. In order to avoid too much deviance of decimals and rounding up or down, the calculation is done by recalculating from a total number of persons, obtained from multiplying the original percentages by 10. The numbers are then rounded to their closest number without decimals.

This gives us:

Low: Equals 429 out of 738 persons = 58 %
Average: Equals 257 out of 738 persons = 35 %
High: Equals 54 out of 738 persons = 7 %

The skill levels can be described as follows. Readers may consider which description and corresponding IT skill level they can best recognise or place themselves in:

Low: Can use applications that they are familiar with for well-defined tasks. This implies that such users will easily be confused, or feel challenged, if or when new applications or interfaces are introduced, or if their applications perform or give results in unexpected ways.

Average: Is ready to use new applications alongside with familiar ones, and perform task such as sorting or monitoring, and is also prepared to occasionally handle unforeseen results. This implies some autonomy, where such actions as actively obtaining and installing new applications and taking them in use is something that people in this category will do as needed.

High: Is prepared to perform complex tasks with familiar as well as unfamiliar applications for workflows that may demand decisions on data sorting, workflow adjustments, etc. This type of users will often solve problems on their own by combining steps – such as search for advice and apply it, e.g. by changing advanced program settings.

Please remember that the levels are generalisations, and that there are no fixed boundaries between them.

It would be misguided and contrary to the point to attempt to define a clear distinction between, e.g. someone at the higher end of the “low” category and someone at the lower end of the “average” category. People within the categories can have different skill levels, and the categories will also overlap with their neighbouring ones.

The major skill IT levels that likely characterise scholars and students in the humanities may now be illustrated – roughly – like this[4], with taglines that are not absolutes:

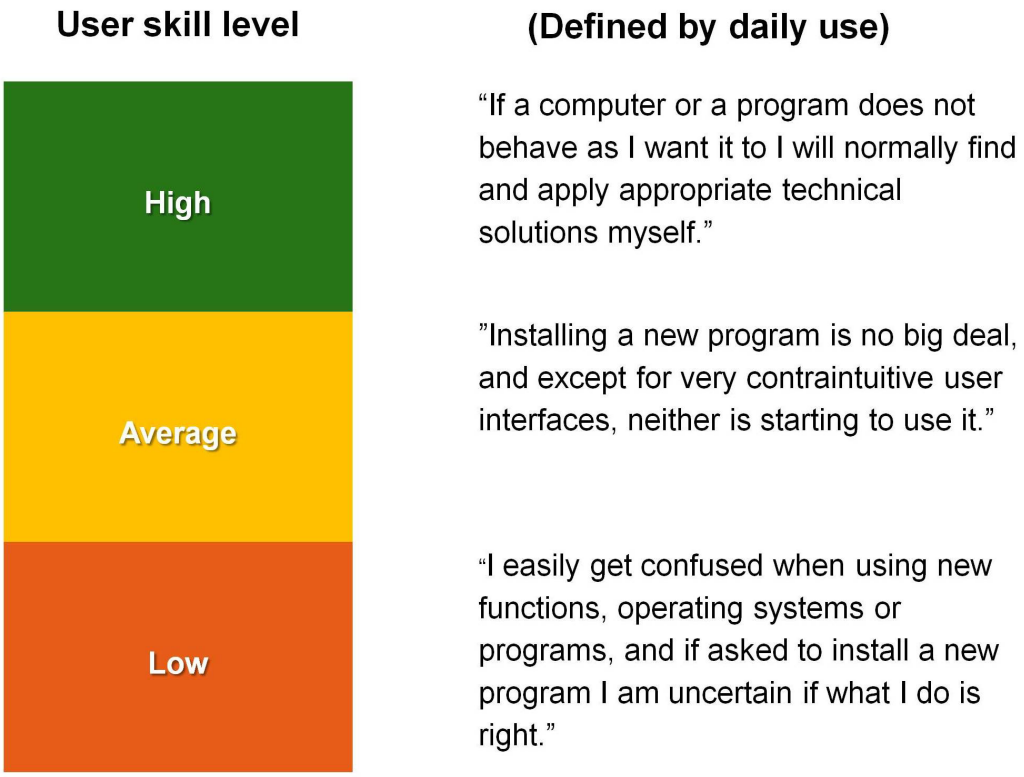


Figure 1: Skill levels by daily use.

Recalling Jane Winter's words cited at the start of this chapter, that

"working with web archives (…) requires skills that many historians do not have, and in the short term may be unwilling to learn; it involves acknowledging a degree of ignorance with which otherwise seasoned researchers may be uncomfortable (Winters 2017, p. 2)"

– readers who may best recognise themselves e.g. in the "low" category can take heart, that:

[4] Powerpoint illustrations from Asger Harlung: "IT Proficiency - for Scholars in the Humanities, Their Projects, and Their Knowledge Work", a guest lecture given at The University of Southern Denmark, May 5, 2021.

- You are not alone, but actually in good company. Almost two thirds of your colleagues will probably recognise themselves in the same category.

- Being in the low category is not shameful. It reflects interests that are not especially focused on IT, and it also reflects that you are used to specific types of use that are probably both meaningful and cover what you have normally been needing for your work.

- You are certainly not “cut off” from using web archives, or saving online content that you might need stable copies of. But you will need to consider ways of including stronger IT skills for projects that may require large-scale archiving and/or analysis of such content.

In order to work with web content, online or archived, readers in any of the categories will specifically need at least a basic understanding of the underlying technologies of the Web and web archiving, something which the chapters 4 The Web as a Technology, and 5 The Web Archiving Process should hopefully provide.

3.4 Relating IT Skills to Project Complexity

Research projects involving archived web can have very different topics or goals, with complexity depending on research questions, the type of data needed, the amounts of data needed, where, how, and to which extent the data can be obtained, etc.

In other words, the complexity of a project cannot be defined from fixed criteria. One must instead rely on what is known, such as the amount of data needed, how the data may be found, the levels of data cleaning and preparation that may be needed, and of course the intended type(s) of analyses.

If someone wants to analyse and discuss the content of a limited number of articles, it is only necessary to have the articles in a stable

form. For web content the complexity here is limited to either finding copies in an archive, and/or saving copies of the relevant material oneself.

This is the type of study that literary scholar Franco Moretti calls “close reading”, where a limited number of artefacts are studied and analysed, and possibly compared and discussed against each other. (Moretti 2000, p. 56-58)

With archived web content such an undertaking may prove less easy than it may first sound, because an understanding of what may be missing and why, or for finding the most relevant copies, or for deciding upon the best way(s) to preserve copies, etc. will be necessary. But the demand on IT skills will not go beyond the need for background knowledge, or possibly the use of a few applications and strategies for saving content oneself.

At the other end of the spectrum one will find projects where large amounts of data need to be retrieved (and possibly extracted and transformed), compared and analysed. This could be thousands of posts on social media, or finding and tracking trends on a specific topic over a number of years, or how a group of large websites such as governmental or news websites have developed, etc.

While close reading of selected artefacts would be possible, and often relevant in such a study, such large amounts of data cannot be treated solely by close reading. Automated searches and analyses will be called for – not to mention that some data cleaning will likely be necessary, e.g. deleting duplicates, correcting text encoding that may result in misinterpreted characters from one encoding to another, etc.

If this sounds technical it is due to the good reason that it is. However, the implication in many cases is only that more applications must be used, some of which may be more advanced than basic applications for saving content, and a greater understanding of the data type(s) and possibilities of treatment may be called for. Some projects may require programming skills or expertise, but many will be doable; just not

without acquiring skills which, again in Jane Winters' words "many [scholars] do not have, and in the short term may be unwilling to learn".

An example of a large scale project that will require a high IT skill level could be a project where data is extracted from a web archive for large-scale computational analysis.

On studies of large amounts of content (big data), with a study of the national Danish web domain as an example, Brügger, Laursen and Nielsen say:

"Studies of a national web domain inevitably move from the close and detailed reading of individual web elements such as images, web pages or websites to what the literary scholar Franco Moretti calls 'distant reading'. This refers to a reading that zooms out from the individual document to encompass a vast amount of texts (Moretti, 2000). The aim of a distant reading is to identify systems, structures, patterns and tendencies that transcend the individual texts, at the expense of complete knowledge about each entity in the mass of texts (Brügger, Laursen & Nielsen 2017, p. 62-63)."

It is worth considering the rewards of collecting research data by automated processes. While it may require new skills to get to the point where the automated analyses become feasible, the promise of having the computers do the hard work and yielding results from vast amounts of data beckons at the end.

The levels of complexity in research projects with archived web content may now be illustrated like this; once again with taglines that are not absolutes:

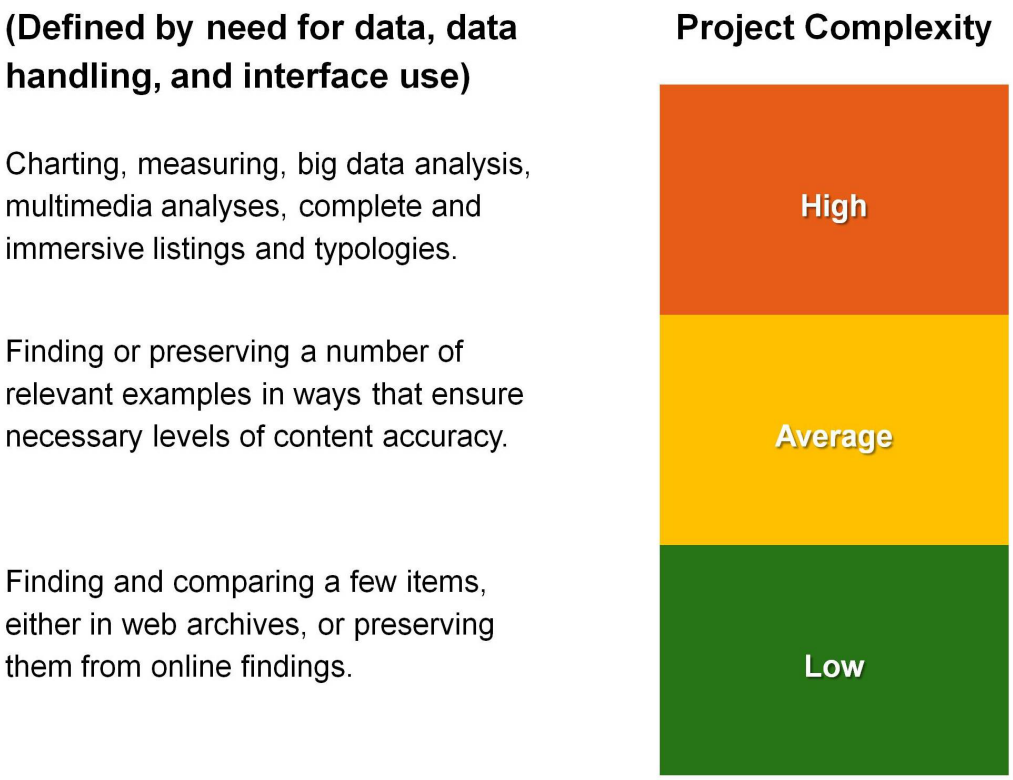


Figure 2: Project complexity estimated by data need, handling, and interface use.

It has been argued so far, that the IT skill level(s) of one or more researcher(s) or student(s) should be considered against the complexity and scope of a project involving web data.

Depending upon one's own estimated IT skill level, and the estimated project complexity, the feasibility level of doing a specific project; or the level of resistance that one might expect, can be illustrated with the following three diagrams:

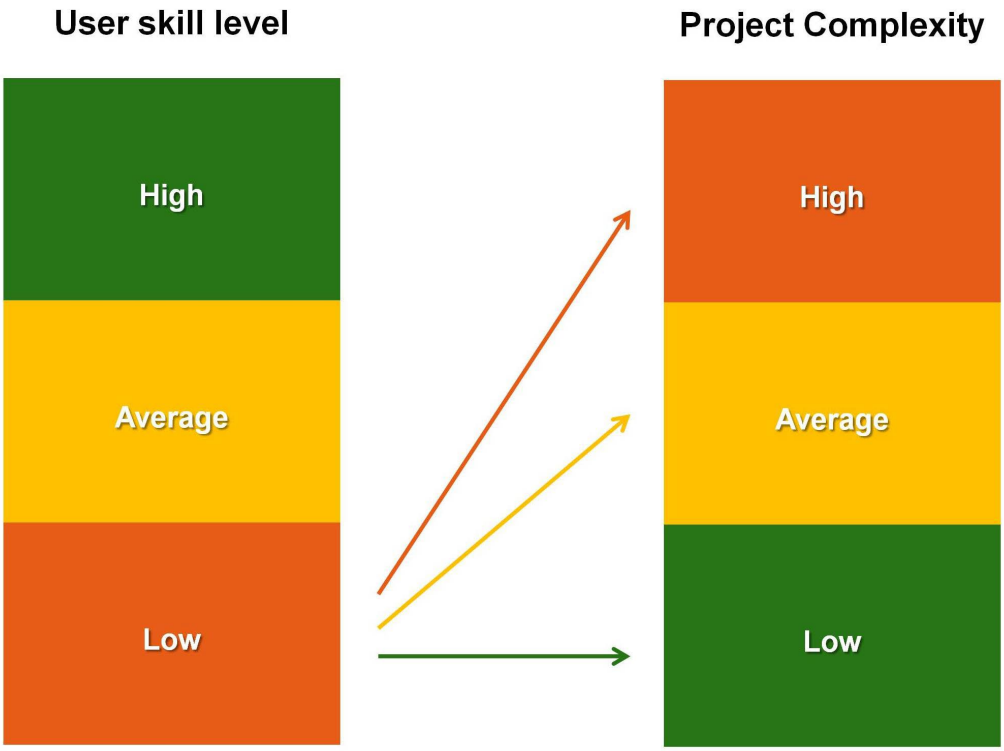


Figure 3: Connecting low skill level with project complexity.

People with low or basic IT skills can safely start out on projects with a corresponding complexity level, indicated by the green arrow.

For larger and more complex projects, the additional demand of needed IT skills must be taken into account. A project of average complexity could imply handling a larger number of artefacts, or a need for some level of computerised analyses that are not highly specialised, e.g. the use of spreadsheets to sort and analyse data.

If the extra skills needed are not excessively high, then a project is feasible, with the understanding that some new skills may be required, and that some challenges may be expected, e.g. in getting the right data or in learning to use new functions or new applications. This is indicated by the yellow arrow.

The red arrow between low IT skill level and high project complexity does not mean “not doable”, but it means that there is a gap that must be resolved, e.g. by learning new and possibly complex skills, or by

involving people with the necessary skills present, or by downgrading the project's size and scope.

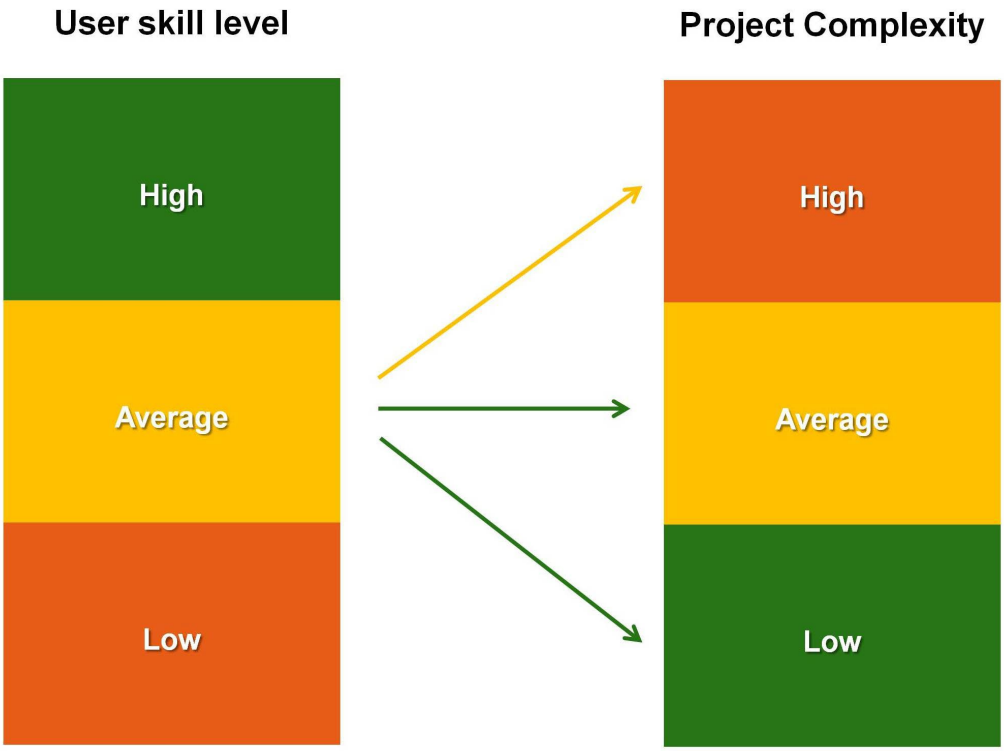


Figure 4: Connecting average skill level with project complexity.

People with an average skills level are rather well prepared, because they are ready to use new programs, and to combine functions for fulfilling tasks. They have almost certainly already done such things on occasion. Therefore, the need for new programs for gathering data, data management or analyses is not an obstacle. Thus, projects of low to average complexity should be accessible to this group.

For projects of high complexity, the researcher(s) or student(s) should try to assess whether the extra demands are for complex data harvesting, handling or analyses which can be handled from applications, with the implications of careful planning and complex data treatment – or if dedicated IT skills such as programming will be needed. The main question here may be, are there programs that are

designed to handle the necessary steps, or will coding in various forms be required?

This assessment may help determine how the project may best be pursued: Should programming skills be learned, if required? Should experts be involved, or can the people presently involved handle the various steps among themselves?

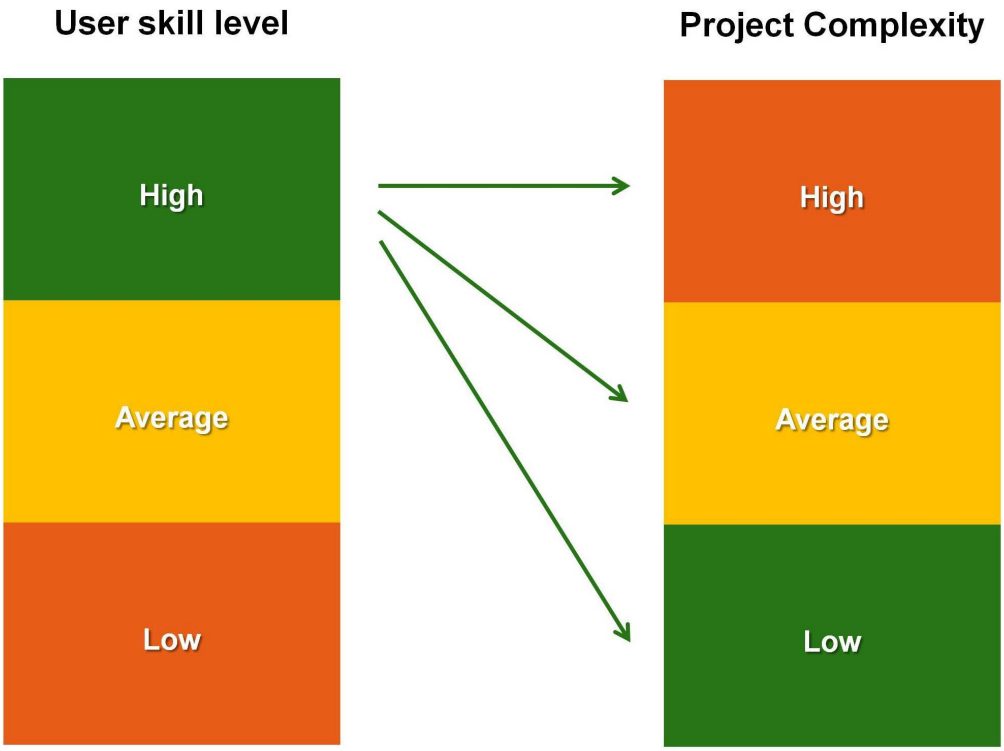


Figure 5: Connecting high skill level with project complexity.

People in the high skill level are prepared for using new applications as well as finding solutions. While complex projects are very likely to call for both this will not be a problem, unless dedicated IT skills such as programming (coding) may be required. It may be within the reach of people in the high skill category to learn such specialised skills, and indeed free online courses in various forms can be found for e.g. Python programming. However, the question of whether specialist skills are called for is a crucial point. Students or researchers may not be inclined to learn such skills themselves, in which case other solutions must be considered.

3.5 Improving One's IT Skills

Some readers may feel that this entire chapter begs the question, "Are there any good ways to improve one's IT skills?"

There is no simple answer to that question, but the mere interest in becoming better is a good start.

Since users are different; in skills, in needs, and in personal attitudes towards IT, no single strategy will fit all. But a few recommendations of starting points and suggested exercises may be in order.

Readers who find themselves in the average or high categories are already well off in respect of IT skills, and those in the average category should just keep on with doing and learning what they need, when they need it, and as relevant.

So the following advice is meant for the largest group in the lower end, who might benefit from moving towards a higher IT skill level.

First of all: Awareness is closely related to skill, and having a language for something equals awareness. The better you can do something, the better you can describe it, and vice versa.

Therefore: Pay attention to names of the programs and services you use. This will often also include file types. One thing is to open your browser in order to visit a web page. Another is to be aware that it is a browser you are opening; that the specific type of program for visiting websites and web pages is "a browser". Yet another thing is to know which browser you are using. For example Google Chrome and Mozilla Firefox are different browsers, and they work differently. The more you know and recognise different programs and file types by name, the more fluent you become, both in describing precisely what you do (which is a highly important point for methodology), and also in finding help and relevant information on your own.

For example: If you cannot see a video on a web page when visiting it in the browser Mozilla Firefox, then you will not find proper help by

searching for “I cannot see a video”, but you will find relevant answers and viable solutions for such a problem by searching for “videos not showing on web pages mozilla firefox”.

Knowing precisely what you are doing and being able to phrase it in words, are key points to the most seamless experience possible with any form of IT usage.

Second: You learn by doing. Becoming better at IT usage is a process driven by a mix of curiosity and persistence. You may sometimes have to fight the impatience that can tempt you to ask others for help with things that you might actually be capable of doing yourself.

Of course, good experiences – successful operations – are much more encouraging and motivating than frustrating ones that ended with giving up. Noone can go from only being capable of heating precooked dinners to being a fairly good cook in just a few simple steps.

So the trick to trying to stand your own ground with IT is to take small steps, trying to do a little more and a little better than you are used to. A good starting point for this could be: If daily functions are basic to friends, family, or colleagues; then decide to try to do what they do, rather than call upon them.

Examples of basic daily functions that many people find practical, but where people with low IT skills will often prefer to have someone else “show them how”, or just simply change a setting for them, could be the following suggested exercises:

1) Change the preferred search engine in a browser, e.g. automatically use Google for searches rather than Bing.
2) Change the start page for a browser – or better yet, set the browser to open where you left off, with the same windows and web pages.
3) Find passwords you may have forgotten but which your browser luckily still remembers.
4) Make the hyperlink and/or menu bar visible in your browser.

5) Make file types (file extensions) visible on your computer.[5]
6) A more elaborate exercise, but not too difficult: Try installing a useful add-on in one of your browsers. This will require you to do more operations: First you have to find out how to look for extensions or add-ons (do a search for this for the browser in question), choose one, and test it. Please refer to subchapter 10.6 Browser Extensions for this exercise.

All these functions are easily changed, but they require that you use the browser as a tool, and not just as a window. Finding recipes for the operations is easy, provided that you know the name of your browser.

For example, try doing a search for “how to start where I left off in Chrome”. Most of the many hits that you will get from this will give you the same and useful directions. If one article is for some reason not quite to your satisfaction, try opening another one of the hits.

Follow the steps described as solutions one by one, and you will soon have made the desired change. And in the process, hopefully found that “I can actually do this”. Some attempts may fail, and end with calling for help. Then the challenge is not to fall back to a less explorative mode, but instead to try keeping on at doing changes yourself; constantly trying to see if you can solve this or that on your own.

Thirdly: Be as explorative as possible. Try to find, and learn to use, new programs that may be useful for you. Hopefully this will provide the double good experience of becoming better with IT while getting useful results in the process.

[5] Common file extensions such as .exe, .docx, .pdf, .jpg, etc. are hidden by default in most operative systems. Having them visible has advantages for security reasons – a file masking as one type while actually being another can be malicious – and it also serves to support awareness of different file types for various forms of e.g. documents or images. File extensions are hidden by default in order to make operative systems look more simple and elegant. So the exercise here is simply to find out how to make them visible, and then possibly hide them again until visibility becomes desirable or relevant.

It is strongly recommended to install one or more alternative browsers if you do not have that already. Pay attention to which browser is which, and what their names are. Consider using different browsers for different purposes, and also consider if a web page does not look or work properly, to try it in another browser. Different browsers handle web pages in different ways, and what does not work well in one browser will sometimes work in another.

Since you are presently reading a book on web archives and web archiving, it is also recommended to try some basic web archiving for yourself. Please refer to chapter 9 Making Your Own Archive, with a special recommendation of the subchapters 9.2 Basic Archiving, Single Pages for exercise purposes. Try to do some or all of the operations described there.

Also consider installing and trying applications that do not look too demanding. Saving web pages, entire websites, or videos found on web pages can be useful and practical in many ways. For example, a saved YouTube video can be viewed without advertisement interruptions, and a saved website can be useful if travelling without constant or stable Internet access.

Basic operations of “doing your own web archiving” are by the way also recommended as a means to get a better feel and understanding of the nature of the archived content, and how it differs from the live content.

4 The Web as a Technology

Takeaways

♦ The HTTP and HTTPS protocols facilitate how a computer may contact another computer and receive the data necessary to show a web page.

♦ The most proper term for a Web address or link is a “URL”, which specifies the protocol, the domain where the content is hosted, and if applicable the subcontent of interest.

♦ URLs strongly resemble folder paths as shown in computer interfaces, and the architecture of websites resembles folder systems.

♦ HTML coding is the core content of all web pages, containing the text along with specifications for all elements on a page.

♦ Browsers are programs used to render HTML code and files into web pages as they are intended to be seen by the user.

In order to understand data derived, extracted, or preserved from the Web, one must first have a background understanding of what is going on when looking at content on the live and online Web.

This chapter will present the basic terms and technologies. This background understanding will also be necessary and relevant when archiving content or working with archived content.

4.1 The Four Fundamental Technologies of the Web

The Internet is a global network of interconnected computers that may share data in various forms, and for many various purposes. It is important not to confuse the Web with the internet, since it is only a part thereof; but an important part used by people globally every day. When someone is looking at a website it is not incorrect to say that he or she is using the Internet; but more correctly the user is visiting “the Web” which is only a part of “the Internet”.

After the prerequisite of connecting computers worldwide by cable, radio and satellites, the Web relies on four primary technologies:

- The HTTP protocol (a connection protocol for computers),
- URLs (unique addresses on the network),
- HTML (the primary programming language for web pages and websites), and
- Browsers (the programs that computer users use for visiting websites and web pages).

The Web works for users without any requirement of understanding any of these technologies, but while researchers or students are not required to have a full and fluent expertise either, basic understanding and knowledge will be helpful, and sometimes necessary for digging deeper for getting data, or deciding on strategies for analysis.

We will need to take a look at what is happening when someone visits a website, but a presentation of the four basic technologies is a prerequisite for doing that.

4.1.1 HTTP: HyperText Transfer Protocol

HTTP is a computer protocol (a set of rules for programming, exchanging and formatting data) which allows computers to exchange the information necessary for users to access websites and web pages.

It is a dedicated type of connection, and also exists in a secure connection form, HTTPS, where the S stands for secure. The HTTPS type of connection is a HTTP connection with encryption of the data transfers in order to be more difficult for third parties to monitor or spy upon.

While it is crucial for computers to understand how data should be exchanged and handled, it is less important for users (programmers and technology specialists excepted).

What is important is to know what HTTP/HTTPS is, and how and why it is a part of URLs that point to resources such as websites, web pages, documents, images, and other files on the web.

4.1.2 URLs: Uniform Resource Locators

“URL” stands for Uniform Resource Locator. It is an address pointing to a specific resource on the Web.

A URL is often referred to as “a link” or “an internet address”. Both of these terms are correct, but they also have broader meanings. For example, a link can also point to a specific page or section in a document, and an internet address can also be an address for something that is not directly accessible for human visitors on the Web, such as a data connection between research or telecommunication computers.

URLs are structured by protocol, main address, and sub address(es) after the following principles:

Figure 6: URL structure overview.

A detailed explanation of each element follows here. It may be helpful for understanding website structures, and in connection with the need for precise URLs for web harvesting. Depending on interests and preferences readers may skip the following explanatory breakdown and focus on chapter 9.3 Copying URLs which provides practical advice for handling URLs when working with archived web and web archiving.

A URL consists of a protocol specification, which in the case of websites and web pages will be either HTTP, or HTTPS, followed by :// which is a computer addressing syntax:

HTTP://

In many URLs the letters “www.” follow just after the “HTTP://”. This specifically means that the domain call is for a WWW server, dedicated to website content. It stems from the early days of the Web, where different services such as websites, emails, and file transfers were spread out over different servers on the same domain, and it was necessary to include the “www.” in order to get the correct server. This is no longer necessary, because domain servers have developed to a point where all the services can be handled by the same servers. After this development, some domains still include the “www.” prefix, while others do not. Modern web browsers will automatically include the

“www.” if it belongs in the exact address, wherefore users do not have to enter it. However, for archiving purposes it is still important to use the formally correct URL, and in this case include or omit the “www” as necessary. Please refer to chapter 9.3 Copying URLs for more information on this.

After HTTP:// follows a web domain. This will consist of a top domain, a domain, and sometimes also a subdomain – in reverse order, and separated by dots (or full stops):

HTTP://subdomain.domain.topdomain
(or HTTP://www.subdomain.domain.topdomain). In the rest of this chapter, the question of the “www.” being there or not is implied, and further examples are omitted.

Top domains are large groups of main domain types with a primary purpose, such as .com which originally stands for “commercial domain”, .org which originally stands for “organisation domain”, .dk which stands for “domain for the country Denmark”, etc.

While top domains are defined from a general, overall purpose, few are strictly restricted to said purpose. A domain such as .gov (which stands for “governmental”) is restricted to governmental institutions in the US, but most domains are accessible for use by any persons or organisations. A single person can buy a .org address; .com is the most used top domain for any website in the US whether or not it is used for commerce, and anyone can buy a “Danish” web address ending with .dk, and create a website in another language than Danish. But the top domain does place a specific domain in a larger category of domains.

Domain is the name of the website in question. It may reflect a service, an organisation, a person, a topic, etc. For example, in “google.com”, “.com” is the top domain, and “google” is the domain.

A subdomain is an additional domain that represents a specific section of a domain, and will only occur in a URL if applicable. Websites of any kind may be divided into special sections represented by subdomains.

For example; the address for Aarhus University, Denmark is https://www.au.dk/, where ".dk" is the top domain, and "au" is the domain. The direct address for the Faculty of Arts at Aarhus University is https://arts.au.dk/, where "arts" is the subdomain, and the School of Communication and Culture has the address, https://cc.au.dk/, where "cc" is the subdomain.

If we look at the tree structure of a website, it follows the same structure as a folder structure, with a main folder, that may contain subfolders, which may again contain subfolders, etc.

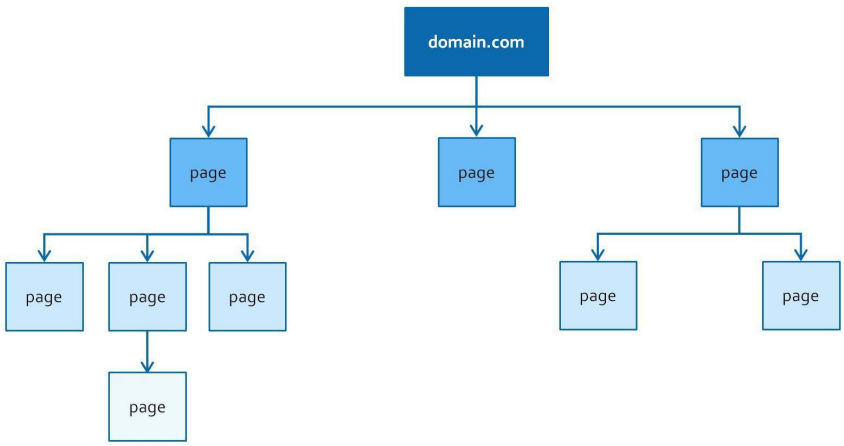


Figure 7: A website structure resembles a folder structure.

The top level contains the "home" or "main" page, in the form of all the files and information that belongs to this page. The levels below act as subfolders with the content for the subpages.

In the URL structure, a subpage is addressed as a file or subfolder would be:

HTTP://subdomain.domain.topdomain/subpage

…and normally, the subpage will be visible from the main page, just as a subfolder will normally be visible in a folder.

In comparison, URLs look and function very much like folder addresses on a computer, as one may see from an example of a folder path in Windows, such as:

C:\My Web Sites\CFI\cfi.au.dk\about

This path specifies the path to a subfolder in a main folder called “My Web Sites”. This folder contains a subfolder (an archiving project folder) called CFI, in which there is a subfolder called cfi.au.dk, in which there is a subfolder called “about”. The path points to the latter. In fact the folder “cfi.au.dk” contains an archived copy of a website with that address, and the folder called “about” contains the content for the subpage “About” from that website.

On a website, the subpage will normally be visible as a menu entry. A subfolder – from here let us go back to saying “page” – in a website structure may be categorised as a subdomain, sending the visitor directly to a specific subpage that serves as an entire separate section, as exemplified above with the address for the Faculty of Arts at Aarhus University; https://arts.au.dk/, and as already stated subdomains go to the front in a URL. But in most cases, the subpages will be addressed as shown in the general example above;

HTTP://subdomain.domain.topdomain/subpage,

– and subpages to a subpage will be added after the subpage they belong under:

HTTP://subdomain.domain.topdomain/subpage/subpage

Just as folders can be pure placeholders for subfolders, a subpage can serve as a path without having any content of “its own”. A very typical example of this would be if a website has a primary language, but offers versions in one or more other languages as alternatives.

In this case one may imagine a folder structure, where all folders and subfolders are copied in translated versions into a separate subfolder. On the main page a visitor may now choose a different language, and be led to the alternative versions of the web pages. In this case the language folder serves literally as a folder with subfolders, as a path that does not lead to a separate page but to alternative versions of the main content.

For example, if one goes from the Danish language “about” page for the Faculty of Arts at Aarhus University which is called called “Om Arts” and has this URL:

https://arts.au.dk/om-arts/om-arts

– then for the English version, the entire URL shifts to:

https://arts.au.dk/en/about-arts/the-faculty-of-arts

– where “/en/” serves as a path to the alternative language versions of the content. Therefore, the formal logical expression of a precise Web address in the form of a URL may have subsections that are categorised as either subdomains or paths, but all the single elements represent structure levels for the website where the content exists.

This is illustrated in figure 8.

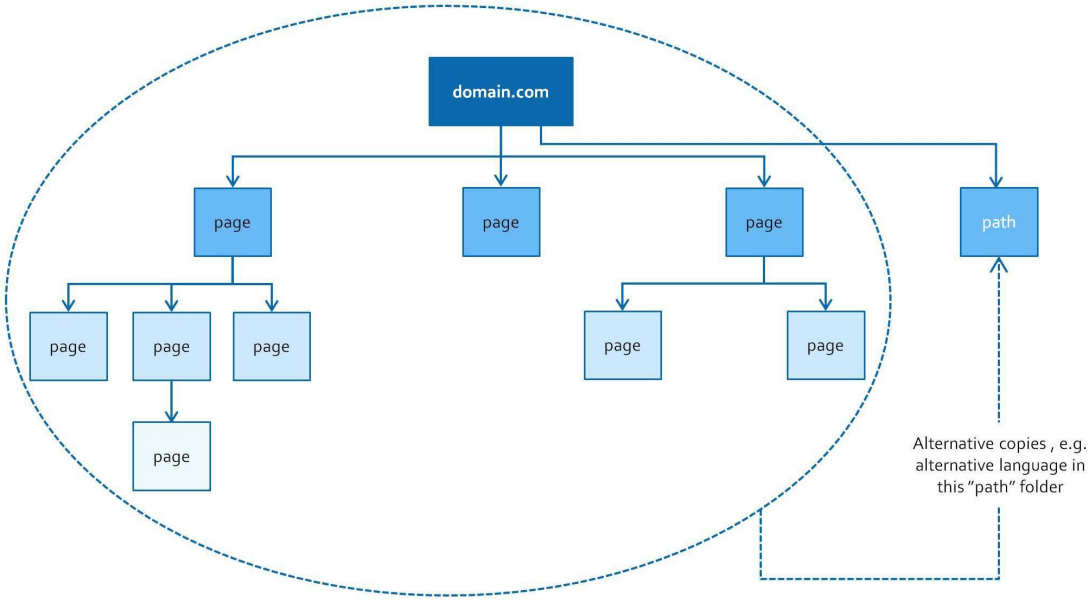


Figure 8: A subpage or subfolder may serve as a path to supplementary or alternative content.

In addition to the primary URL structure, a URL may sometimes contain tracking information. This may contain information on how or where the visitor found the resource (web page), e.g. if a link was followed from a social service, a newsletter, another website.

The tracking information is an addition to the URL, not an integrated part of it. It can disturb searches and archiving attempts, and should be removed for these purposes. Please refer to chapter 9.3 Copying URLs.

4.1.3 HTML: HyperText Markup Language

HTML is a basic programming language for creating web pages and websites. HTML programming relies on HTML tags which specify the types of content, and how the content should be represented.

For example, the tags may specify that a part of a text is a header, or that some words in the text should be in bold letters. Other tags can specify other elements, e.g. that an image file should be placed in the

text, where it should be placed, how large it should be relative to the text, etc. Other tags again specify that something is a link, e.g. to another page, another website, a file, or another section of the present text. Websites and web pages are constructed from text and files, which are primarily controlled and structured by the HTML language.

Exceptions from this are scripts, or embeds where HTML codes are used to call remote computers and/or programs which control specific content in other ways. For further explanation, please refer to 5.1 Challenges for Web Crawlers, p. 77-82 on scripts and embeds. But while they cannot always be copied and archived, such special functions and services are still called with HTML commands/tags. The HTML codes can thus be saved, whereby the content can sometimes be identified, or partly identified. This implies a risk of embeds on archived pages calling in content from the live web which in reality is not preserved in the archive. See 7.5 The Risk of Online Leaking.

The basic understanding that is needed when working with web archives or web archiving is simply, that the core content of a web page is always an HTML file. This file contains the text represented on the page, along with specifications of how the text should look, whether the parts of the text are headings, regular paragraphs, if some words should be bold, etc. In other words, the HTML coding specifies the formatting and layout for the text.

The HTML file also details which other elements should be represented on the web page, such as images, file downloads, document previews, videos, etc. The elements that are not included directly in the HTML file as text may be from the same domain as the one where the web page is hosted, or they may be fetched from entirely different websites or services.

A web page with any other elements than text will thus consist of an HTML file, plus additional files which the HTML code specifies as elements on the page. The HTML code for a web page is also referred to as the “source code”.

The HTML file for a web page is both machine and human readable; a computer or a person can find text or references for specific types of content in it. Such things are not necessary for a person visiting a web page, live or archived, in order to study the content – but it allows for various types of in-depth study:

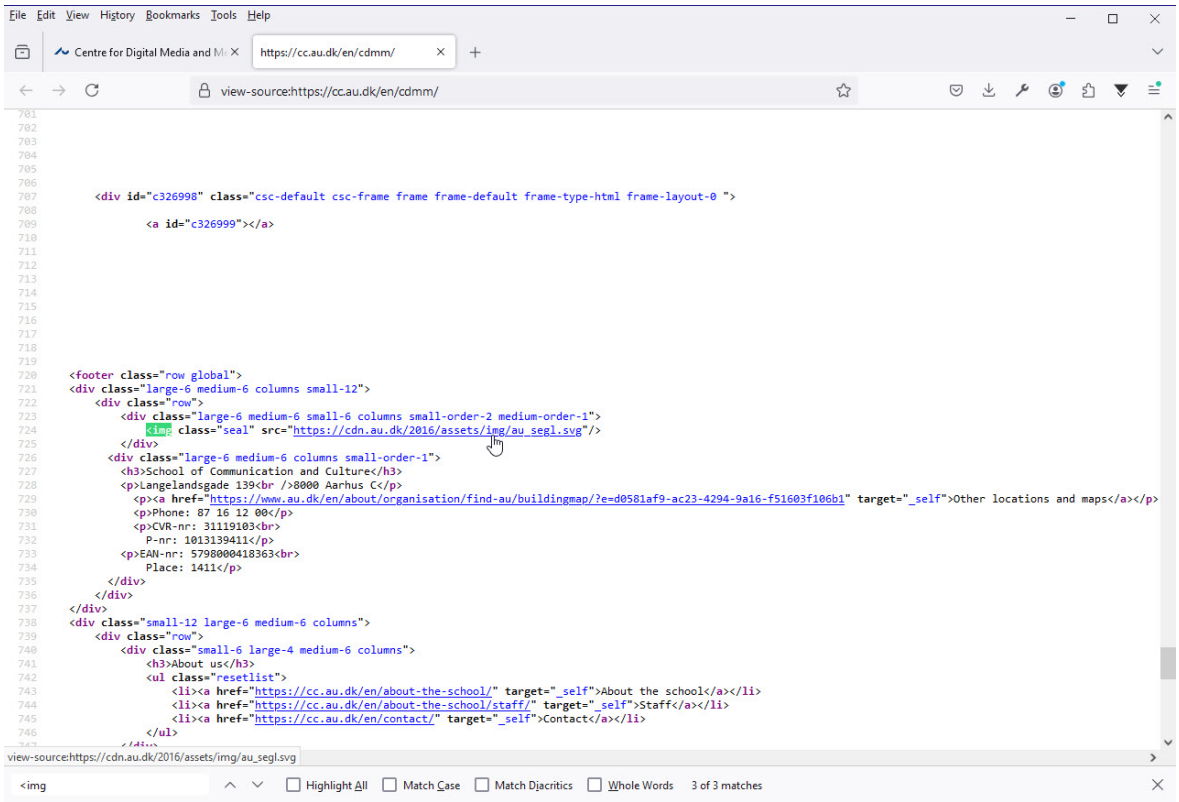


Figure 9: The HTML source code for a web page. Full example given at the end of this subchapter.

A human reader can identify the specific HTML tag for images, or external links, or other types of content, and for example count the number of links a website or web page has to external resources that are not hosted on the website the page resides on, or look for information on content that should have been represented but is missing. Looking into the HTML code can be done both with archived web pages or pages on the live Web. Digging into the underlying code in order to acquire knowledge about hidden or lost content is called “data mining”.

This does not imply that readers must become familiar with HTML codes, but in some cases it can be useful as a way to go deeper in a close reading of specific content. As needed, a reader without in-depth

knowledge of HTML can search for and find the specific codes signifying that something from a web page should be, e.g. an image (<img) or an external reference to another web page (<a href), or an embed of some kind of external content (<embed).[6]

It should be noted here that the full syntax for an HTML tag includes angle brackets at both ends. A search for the proper tags for HTML content will yield results such as “<img>”, or “<a href>”. However, in source code the ending bracket will be placed after the data for specific images, links, or other content types. Since each HTML tag is a command, a search through source code for specific content should look for places where the relevant command starts. Thus, in order to find images or links one must omit the ending bracket and search the source code for “<img” or “<a href”, not for “<img>” or “<a href>”. The latter will not yield any proper results.

Searching a web page for specific types of content via HTML tags, or inspecting the source code for other reasons, requires the user to look at the underlying code rather than the page as is it is normally meant to be seen. All web browsers allow this, but the way may differ, not just from browser to browser, but also over time in different versions of the same browser. Ways of finding and studying the HTML code will normally be identified as functions that will show the source code or allow the user to “inspect” elements on a web page.

Common ways of finding these functions are by right-clicking on a web page (ctrl+click in Mac OS) and look for functions like “View page source”, “Source”, or “Inspect”. Sometimes “Inspect element” will appear if right-clicking on a specific type of content on a web page. In other cases such functions can be accessed from the menu in the browser, provided that is has a menu and that the menu bar is not hidden. Due to the ongoing changes in browsers, when using a specific browser for one’s studies the reader is advised to search for “how to view source code in [browser name]”. This will provide useful hits, often from the developer’s own web pages.

[6] E.g. by doing a search for “html code image”, “html code link”, or “html code embed”.

The following example of looking into the source (HTML) code for a web page shows, that by right-clicking in a 2024 version of the browser Mozilla Firefox Developer one may choose “View Page Source”.

When clicking “View Page Source” a new version with the page shown in its pure HTML form appears in a new tab, as shown in the next two figures.

As a further example, the page with the source code has been searched for instances of the HTML tag “<img”. The highlighted example is the logo for Aarhus University found at the top right of the page in the original view (Figure 10). The hand cursor is placed at the exact URL for this element in the source code (Figure 11).

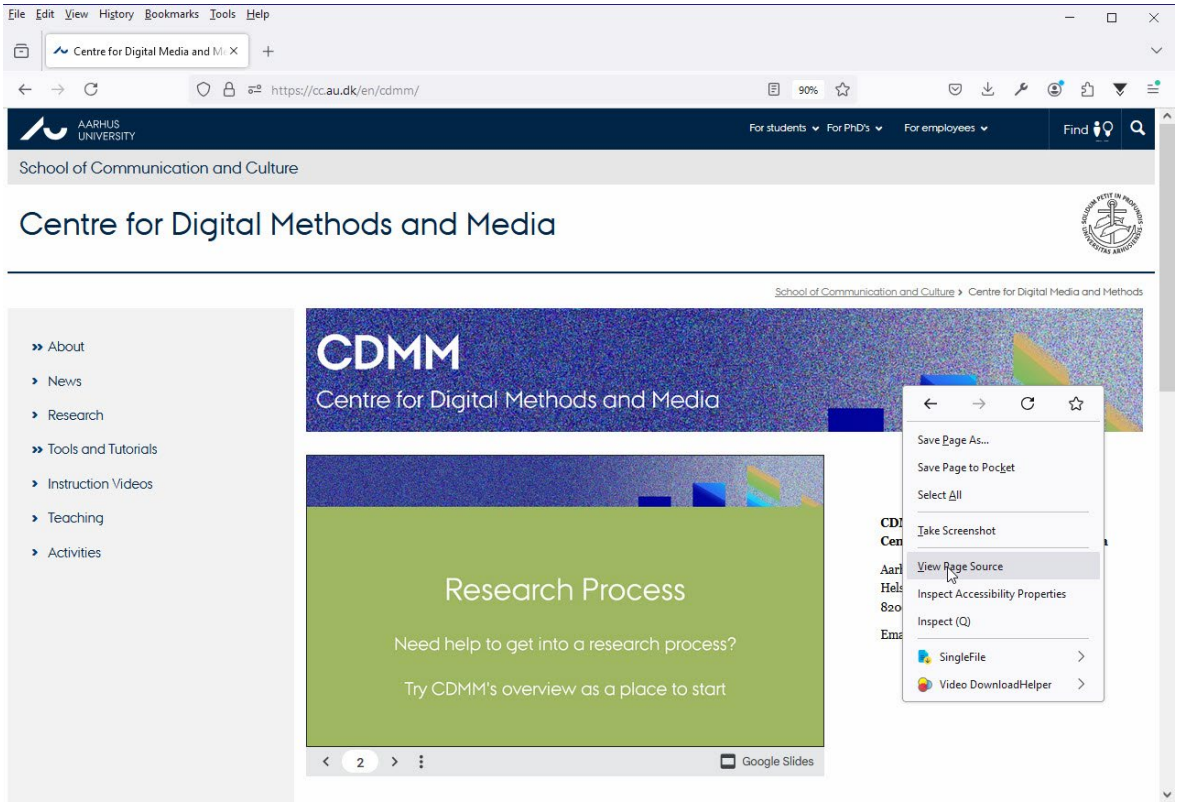


Figure 10: Finding the page source in a browser, example.

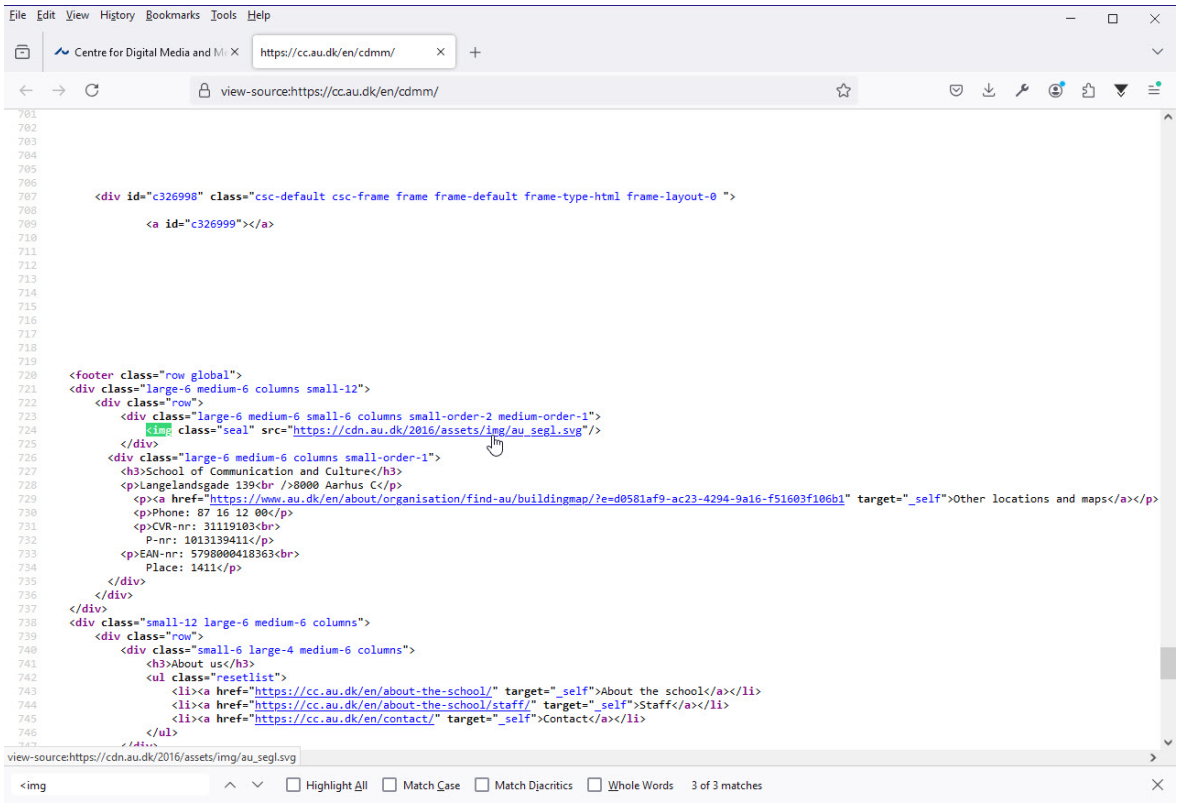


Figure 11: A view of the source code for a web page, example. An image tag is found and highlighted.

Except for very specific cases of close study, readers may never have to look at source code themselves. This is because one of the potentials of the source code is its machine readability. For example, web harvesters (computer programs for archiving web content, see 5 The Web Archiving Process) can be set to find and save all images, or document files, or other specific content types, or identify and list the external URLs on web pages or entire websites.

The machine readability thus allows for computational treatment of the underlying HTML code as data, whereby big data treatment of web content becomes possible, allowing for “distant reading” in one’s studies (see 3.4 Relating IT Skills to Project Complexity, p. 40). While such studies would be too time demanding to be possible for humans, computational treatment can allow for finding answers to questions such as, e.g. “how often is a named politician mentioned in a named list of news websites in a specified time period”, “how do conspiracy websites link to other websites”, “what was the average size of images used on websites in Denmark in the year 2007”, etc.

4.1.4 Web Browsers

Web browsers are computer programs that allow humans to access content on the Web in the manner that it is meant to be seen.

The name “browser” refers to the analogy of “pages” for a display of specific web content: Human visitors of websites will “browse” through their “pages” while looking for content or following links from one web page to another, etc.

There are many browsers available to computer users. Some are built into and included with different operative systems; others are alternatives that can be downloaded and installed.

All browsers serve the same purpose: To retrieve HTML code from a specified source placed on a server, a computer on the Internet where others may access content, along with relevant files needed for the content, and render it into a representation meant for the human visitor, a web page. The browser retrieves the data and represents it in the intended human readable form, as the Web medium of websites and web pages.

When the user enters a URL into the browser, by writing it or by clicking it, the browser will contact that address and retrieve the HTML code. As directed by the HTML code, it will also fetch the files and content needed for that web page. This additional content may be placed on different servers from the one where the primary HTML code is retrieved. This is illustrated in figure 12 in the next subchapter, 4.2 The Sum of the Parts: Visiting the Web.

Browsers have subtle differences, not just in their layout and functions but also in the way they specifically handle the data retrieved from a server. A feature on a website may be designed in a way that works well in most browsers, but not in all. Encountering something that does not work, e.g. that a “download button” does lead to a download, does not mean that the browser or the website is flawed, but rather that an unforeseen conflict exists in a specific case. Website developers who want special functions or effects on their web pages may test and

develop until they are sure that the functions will work on the most popular browsers, typically including those that are built in on Mac or Windows systems; but the special functions may not work correctly on browsers that they were not tested in and optimised for. As another example, online banking services will often specify that their customers should use specific browsers; possibly for security reasons, but also possibly because the services may rely on functions that are known to be supported in the recommended browsers.

The fact that the same web pages may not work equally well in any browser is a good reason for having more browsers, and for testing web pages that do not work correctly in one browser in another.

But there can also be subtle differences, e.g. that some browsers will render headings as coloured while other browsers handle the same headings as black, or render stings of numbers of certain lengths as links to a phone service application based on the assumption that the numbers represent a phone number, or allow or block videos from starting to play as soon as a web page loads.

On a side note; as browsers and websites have developed, many old web pages were meant for and tested in older browsers that were programmed to work in older systems, and cannot run on modern computers. These web pages may still be seen in a modern browser, but may not look or work fully as they were intended to.[7]

[7] While this book attempts to avoid referring to services and technologies that may change, it is worth mentioning that older websites can – as of 2024 – be inspected in emulations of their contemporary browsers, e.g. in the service Oldweb.Today; https://oldweb.today/. Oldweb.Today may be blocked on work computers by organisation security because it appears strange, but it is a legitimate service.

4.2 The Sum of the Parts: Visiting the Web

The four primary technologies work together like this:

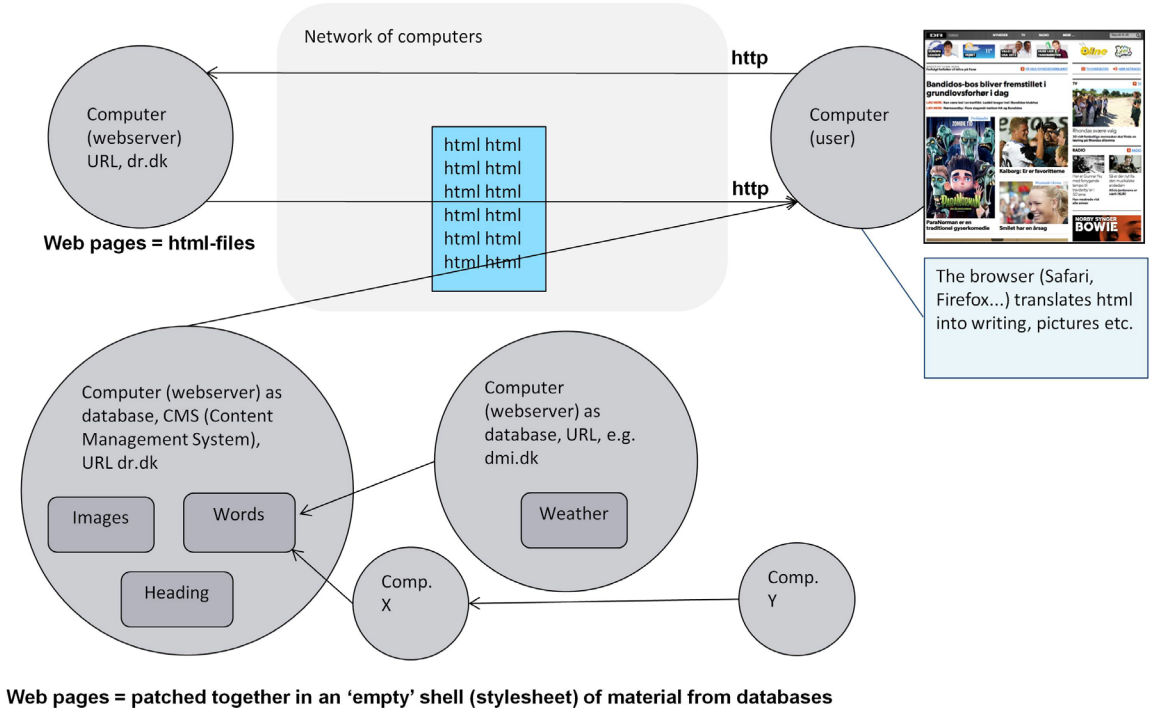


Figure 12: The technology behind a visit to the Web.

Using a browser, a user makes a call to a specific web address (a URL). The browser contacts the specified web server, and retrieves the HTML code that defines that web page, along with some of the files to be used on the page, which will often be found on the same server, but as separate files placed elsewhere in the system, e.g. in an image database in a Content management System (CMS).

It is also very likely that the HTML code calls for external content such as images hosted at a third party (another web server), and services such as videos or animations hosted on yet other servers.

All this is placed and shown in the browser as specified by the HTML code which forms the basis of the web page. The main content comes from the primary server as HTML code, but some of what the user sees in the browser may be content retrieved from other places.

4.3 Observation Units of the Web

Researchers or students in the humanities will normally not be interested in specific technological details behind web pages, but to the extent that they may serve to identify and retrieve content that carries meaning for human recipients.

The smallest units of the Web that carry meaning meant for a human audience are web elements; the various elements used for building web pages, such as main text, images, text boxes, link references (URLs, but possibly presented as links such as “click **here**”, or as an image set to direct to another page), etc.

But depending on the interests for a specific study, the Web can be studied and observed on – or as – several levels of expression and content:

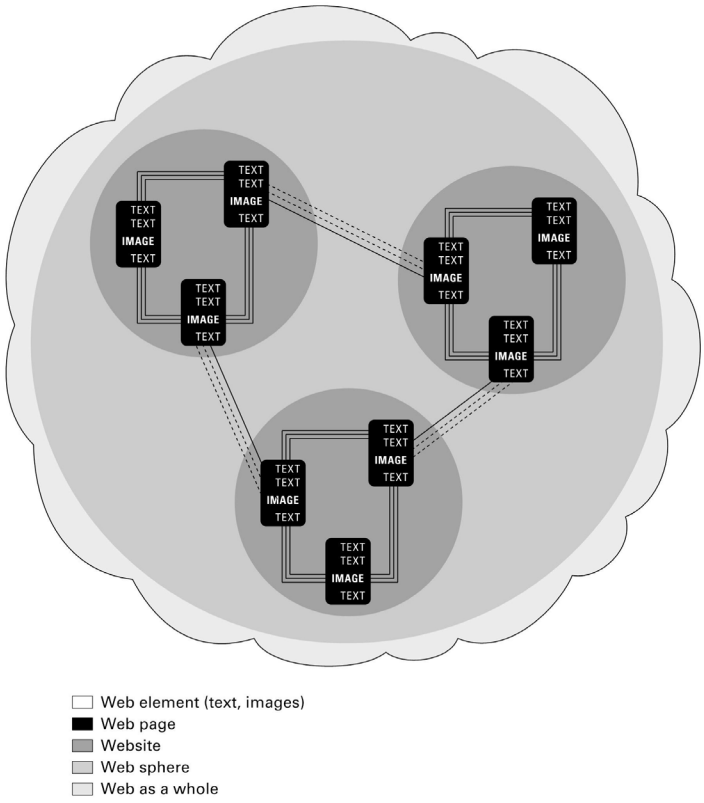


Figure 13: Illustration of the layers or units of the web. Illustration taken from Brügger, 2018, p. 32.

Web elements are parts of web pages, and are meant to serve as parts of a connected whole in the total expression of each web page.

Web pages are parts of websites on which they, too, contribute to the website as a whole.

Websites can be seen as elements in one or more groups of websites that can be regarded as web spheres; for example websites on a specific topic, websites with a shared agenda, business websites overall or business websites for specific areas of business, websites in a specific language, websites on a specific top domain, etc.

And finally, there is the Web as a whole. While extremely big and consisting on amounts of data that cannot be studied or encompassed by any single human or group of humans, examples of studies addressing the Web at this level exist, e.g. the ones mentioned in 1 The Importance of Web Archiving, p. 17 with statistics of how often websites change or disappear.

5 The Web Archiving Process

Takeaways

♦ Web content is archived by “harvesting” it; that is finding and copying it. The process is referred to as “web harvesting” or “web crawling”, and the programs used in the process are referred to as “harvesters” or “crawlers”.

♦ In order to get external content that is hosted on web pages from other domains a web harvest must be allowed to leave the pages intended for the harvest and retrieve content from other pages. This results in extra web pages or even entire websites being harvested although they were not explicitly targeted. This is called a by-harvest.

♦ Websites point around to their pages internally, and harvesters can be directed back and forth to the same pages. If the pages change during the time that the harvest runs, this can result in more copies of the same page in different versions.

♦ Web harvesters can save HTML files and other file types, but there are types of external content that cannot be archived in the process, e.g. social media or streaming content.

♦ Subtle changes happen with the archived content. Apart from the question of some content not being harvested, URLs on web pages are modified to work in the archived copy as “archive URLs”.

Web archiving is done by starting a web harvester, a program dedicated and designed to visit and download web pages systematically. The process of harvesting is also referred to as “crawling”; going from web page to web page and saving content.

The institutional web archives have advanced programs for large scale harvesting, but private persons can also use web harvesters, for more on this please see 9.4 Archiving Websites or Specific Content from Websites.

The structure of a website is divided in levels resembling the folder structure in operative systems as described in 4.1.2 URLs: Uniform Resource Locators.

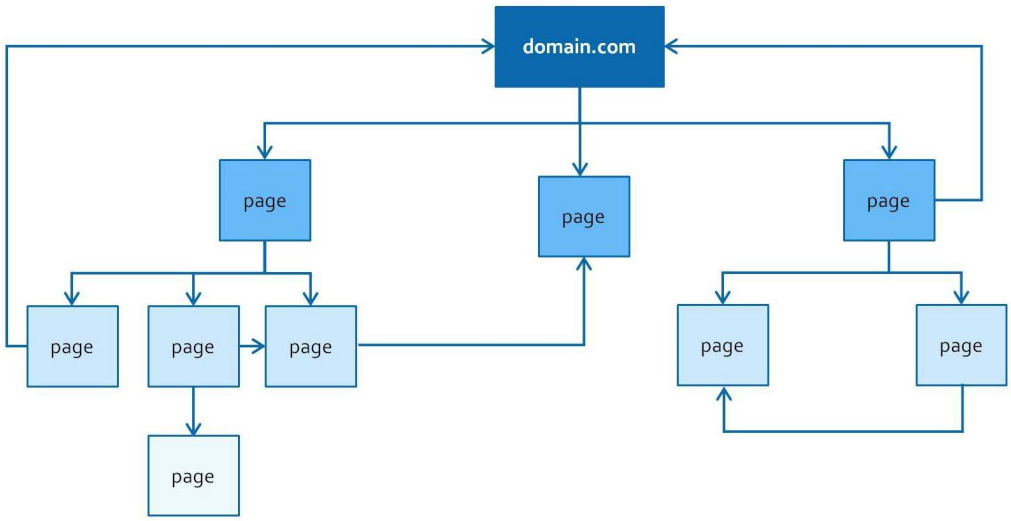


Figure 14: A harvest of a website may be pointed back more than once.

A web crawler will follow the URL for a website, copy the content, find and follow the links for pages linked from the first page, repeat the process with those pages, and so on. This will primarily follow the menu for a website, where the main page, or “homepage”, is the first and entry level, the primary pages or sections in the menu are the second level, the subpages for pages on the second level are the third level, and so on.

However, links to other pages on a website may also appear in other places than the menu; for example as a recommendation of a specific subsection or a reference to another page on the website. This may occur on any page or level of a website.

In theory this can send the crawler to a page that it has already visited, resulting in that page being archived again. In this manner the crawler may repeatedly be directed up and down on the website to pages it has already visited and copied.

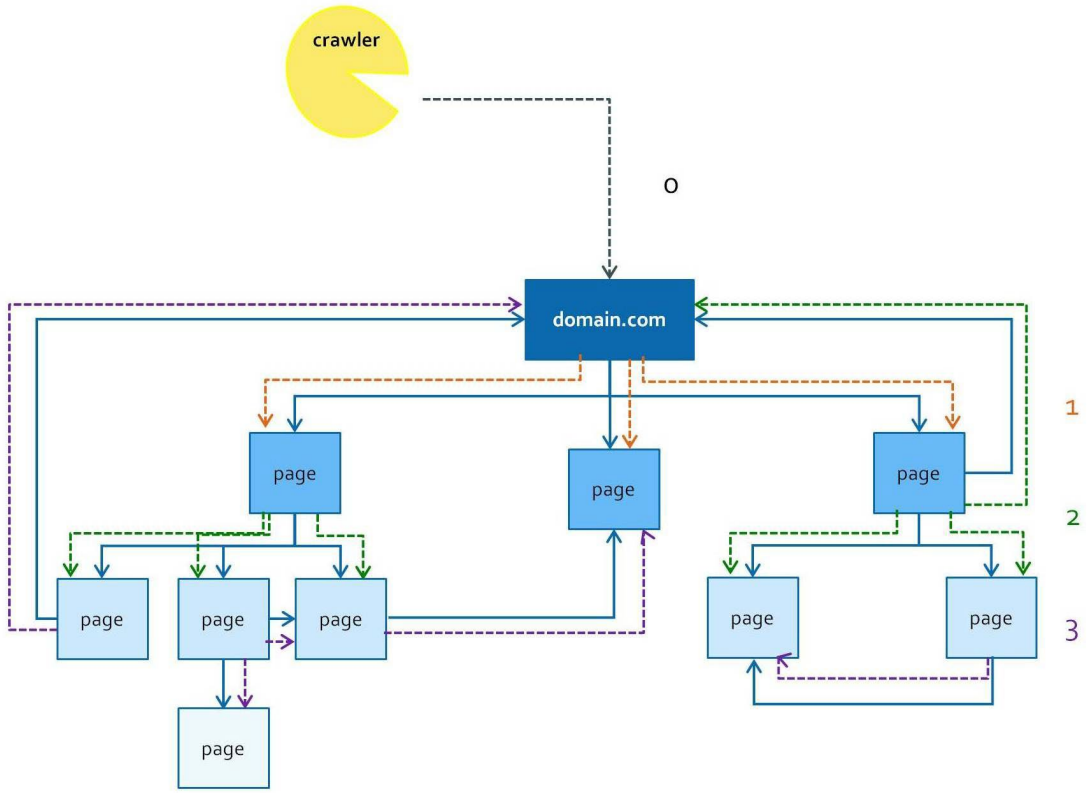


Figure 15: URLs on the pages that the crawler visits can direct it back to pages already crawled.

As crawlers have become more advanced and archivists more experienced, this side effect has been limited so it is more likely to be found in older sections of web archives, and so that re-archiving of the same page in a single harvest job should normally only happen if the page has changed if or when the crawler revisits it.

But the problem of the same pages being archived more than once in a single harvest cannot be ruled out entirely – it is a question of

settings and limitations, and any limitation defined may be at the cost of content that should have been archived.

As for reasons why several copies of the same page can be a problem; when visiting a web archive in order to find a specific web page from a specific date, one may find several copies of that page. They may look exactly alike, or they may have differences, and the web archive visitor may face a challenge in determining, which of the archived versions of that web page is the most proper version for the purpose of the visit.

Another side effect from having many representations of the same content is data redundancies which cause overrepresentations. For example, if a researcher wants to find how many mentions of “fake news” there were on a news website in a specific timeframe, a correct count cannot be made until duplicates have been rooted away.

When starting a job – a specific harvesting process – a seed list is entered into the web crawler. The seed list is a list of specific website URLs that the crawler must attempt to harvest. Seed lists help the web archives to prioritise and strategically maintain the content of the web archive. For example, news websites change rapidly and are therefore harvested frequently, usually several times daily. Websites of high importance such as governmental websites are also harvested frequently, but while it is important to archive changes, the pages are not expected to come and go by the hour, wherefore a harvesting frequency of several times per day is hardly necessary. Websites such as blogs or small businesses could be examples of websites that the archives would want to copy a couple of times annually. All this is controlled by seed list harvesting, used in planning for when and how often specific groups of domains should be harvested.

Other settings will also be entered into the crawler, such as how far away from the seed list, and how deep down in the seed list it is allowed to go, whether to abort copying from websites where the crawler seems to be caught in a loop (e.g. by attempting to copy a

calendar function that endlessly generates new dates), whether to skip files that were copied in a previous crawl, etc.

The settings will affect several things, including the thoroughness, the time required for the job, and the amount of data being harvested.

As described in chapter 4.2 The Sum of the Parts: Visiting the Web, some types of elements for a web page may be services called in from other domains, and may represent various content types that may be files (images, documents, etc.), but can also be content delivered via programs running at the service providing domain. The latter type may not be possible to preserve; it simply is neither HTML nor files and cannot be copied by the web harvester.

File types emulating various media (“mime types”, mimicking known physical media such as images, documents, audio, video, etc.) or programs for download, are often hosted on other domains than those found on a specific seed list. The web crawler cannot retrieve such external content unless it is allowed to follow links that lead outside of the domains on the seed list.

In order to get a high level of completeness of copies, a web harvest will therefore permit the crawler to go a number of steps outside of the target seed list. Assuming that a web page calls in an image from a service provider, the web crawler must be allowed to go to that service provider in order to retrieve it. Without that permission, the copied web page will be missing that image. However, going only one step outside of seed list in order to retrieve content will in many cases not be enough; a service provider for images may have a primary domain for the service, but it may itself retrieve the images from another domain where it is stored, e.g. a content management system (CMS).

When the harvester is allowed to go to external domains it will also encounter URLs on those domains; URLs that were not part of the intended harvest and which may again point to additional external domains. The harvester will pursue them within the delimitation of “how many steps outside of the seed list it has been allowed to go”.

An unlimited number of steps outside of the seed list could theoretically result in an attempt to harvest the entire Web; the harvester will find new links, pursue them, find more new links, pursue them, and will not stop until all links have been pursued and harvested. Therefore, delimitation is necessary for running a feasible harvesting project. But the delimitation cannot be too strict if the intended content is to be harvested in a quality that can be rendered as fairly complete representations.

So when a harvest takes place, domains that were not targeted in the seed list will be included. This is illustrated in figure 16, where the harvester is sent from the first domain on the seed list (blue) to a domain that was not there (violet), and from the second domain (red) to yet another domain outside the seed list (orange).

Furthermore, the unlisted domains may point back to some of the domains that were listed, which can cause the harvester to go back for yet another copy (violet domain sending the harvester to green domain, etc.), which can again result in duplicates of the same pages harvested at different times of the harvest process.

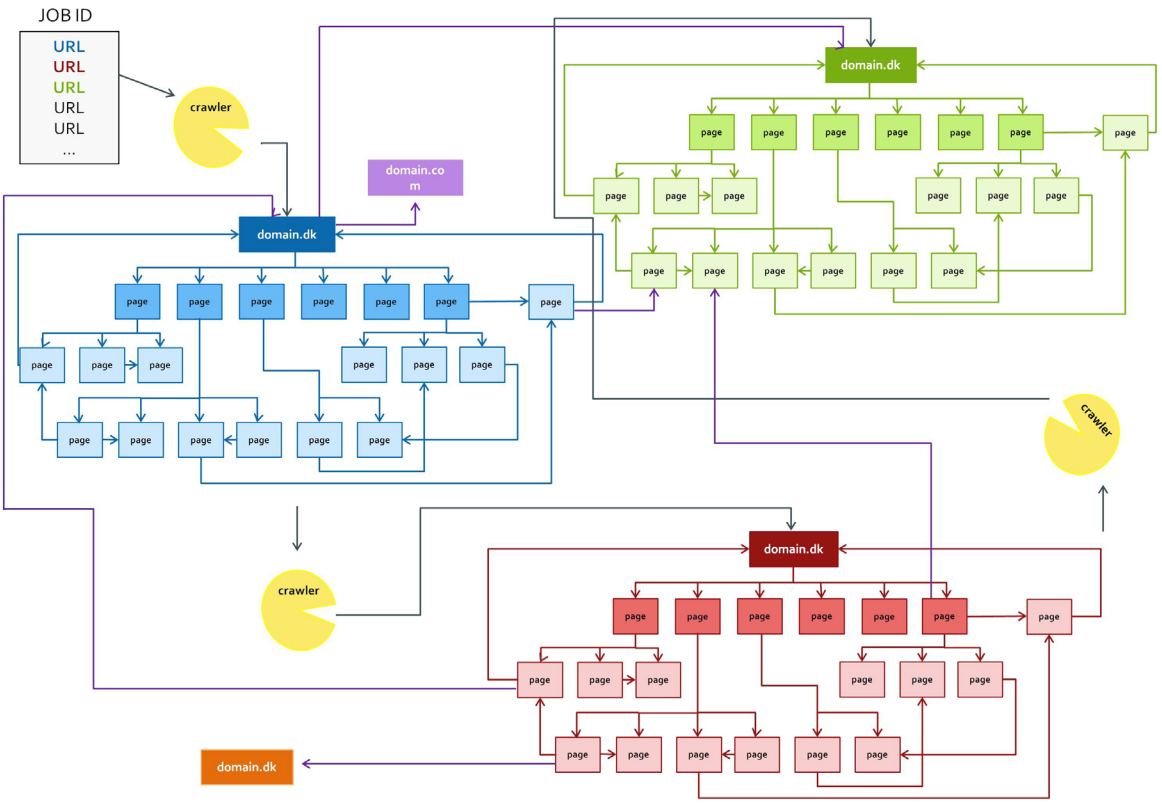


Figure 16: Some domains may point back to domains already harvested.

Over the years, steps and delimitations have been developed to limit the effect of unnecessary double harvesting, but it is not completely avoidable. For this reason one may expect to find more versions of a given page for a given time, often without any kinds of visible differences, in a web archive.

But furthermore, a harvest will include websites and web pages that were not specifically meant for the harvest. This extra data is added to the archive, along with the websites and web pages that were intentionally harvested.

When a new list of domains are given as a seed list for a new harvest job, chances are that some of those domains will point back to domains that were nor specifically targeted, and possibly already harvested in a previous job. This will result in an additional harvest of those domains that were outside of the planned harvesting. The extra copies added to the archive from the various crawls without being

intended as primary targets for harvesting, are by-harvests. By-harvests will occur whenever a crawler attempts to fetch content outside of the domains on the seed list.

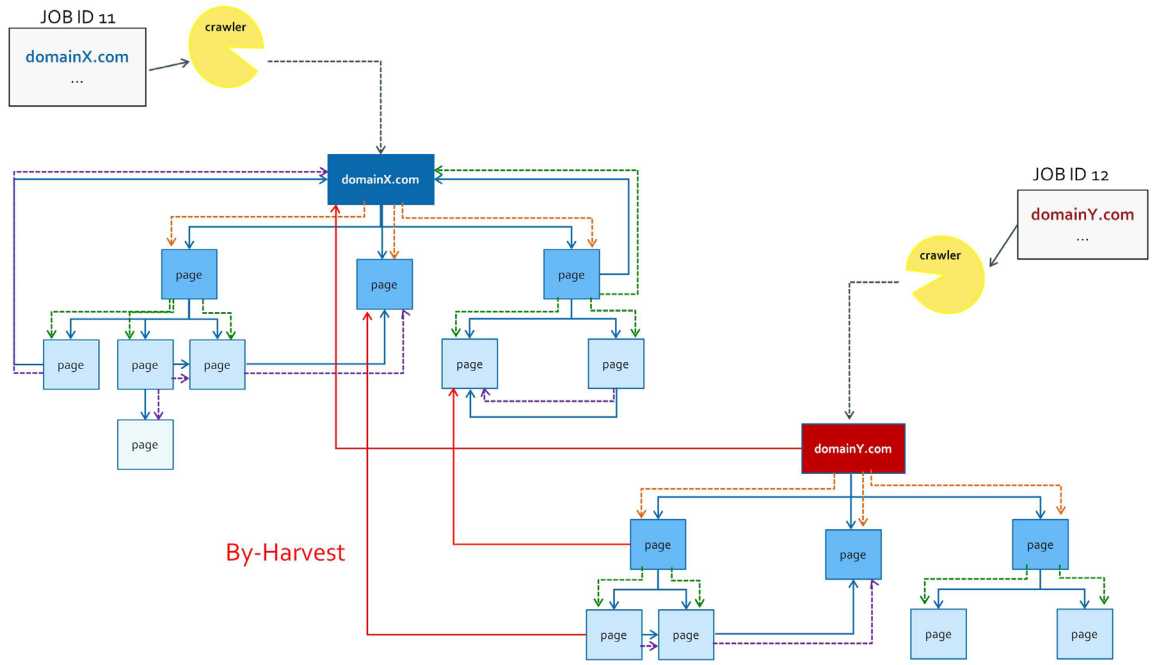


Figure 17: Harvest jobs may cause harvesting of domains that were not specifically targeted, and possibly harvested in previous harvest jobs.

By-harvests have advantages and disadvantages. The advantages are that they add to the chances of finding copies of content that was never planned for harvesting e.g. from hidden web pages or unregistered domains, or extra versions of deliberately harvested content within, or closer to, a specific timeframe.

The disadvantage on the archiving side is that the crawling process will take longer and fetch larger amounts of data. On the user side the by-harvests add to the risk of finding many, sometimes hundreds, of more or less identical results when looking for a specific page. If a search for a page, e.g. in the last half of 2016, results in hundreds hits the user may be able to select a suitable version at random. But just as readily the user may face problems in determining which version or versions to use. Are they all identical? Are they equally complete?

It may be noted here, that searching for content in institutional web archives will provide content chronologically, or alphabetically, but not in any way sorted by relevance, popularity, the user's own search history, or similar, such as users will expect when doing searches on the live Web.

5.1 Challenges for Web Crawlers

Even though the web crawler is harvesting more than it was specifically requested to do in order to ensure the fullest possible content preservation, there is content that cannot be harvested.

Primary challenges that can or may cause some content to not being harvested are listed here:

Robots.txt
Robots.txt is the standard name of a small text file with which website owners can specify that they want to discourage automated access to, or copying of, their content. Such a request is *not* legally binding, and can have various reasons, e.g. that a website has been made for friends and family in connection with a private occasion, that it contains content that cannot be copied, that it has content that is misleading or dishonest and the website owner does not want proof to be saved, etc.

A web harvester is in the category of programs referred to as robots, or "bots", since they act as automated entities designed to perform jobs "on their own and as requested".

Captchas
Captchas (Completely Automated Public Turing test to tell Computers and Humans Apart) are small tests designed to be easily resolved by humans but difficult to computers. They may consist of blurred or distorted letters which the human visitor must recognise and enter, but which a program will have difficulty in interpreting correctly, or in small puzzles based on image recognition, e.g. click the correct motives in a selection of images, or solve a simple puzzle.

Captchas are designed to prevent automated programs from accessing websites and services; primarily as a means of protection against various forms of unauthorised access, e.g. by hacker programs searching for security weaknesses that may be abused, or bots attempting to place messages such as spam or scam advertising or fake news.

Web harvesters face the same problem as other types of programs; they cannot solve a captcha, and therefore cannot access the content that it protects.

Account-based Access

Websites that demand a user registration or a subscription in order to access content cannot be harvested unless an account is specifically created for the harvester, or access for the web crawler is provided by the website owner, e.g. by allowing access for the IP address doing the archiving.

The latter is normally the case with news websites such as newspapers, where an agreement has been made between the web archive and the medium. Owners of paid content will not be willing to have copies freely accessible via open web archives, but after a time when the news are no longer news they, or some categories, may become fully accessible. This will depend on the type of content and on the agreements made.

Websites that are less important than news sites may not be identified or get the extra attention of creating accounts specifically for harvesting, and websites where the content is copyrighted and intended to remain behind a payment wall will normally not be harvested to or accessible in web archives in a complete form.

For example, for journals or magazines the user will normally be able to find the preview that non-paying users would have been able to see on the live Web, whereas full articles cannot be accessed.

Interactive content or scripts
Some websites are “dynamic websites” where the content performs and responds to the user’s behaviour; for example business sites where special offers may be generated or advisory chatbots can appear in response to the user’s actions, or a web page with a virtual piano where the user can play a melody by pressing (clicking) the keys. These types of content are interactive, created and controlled by programming on the server side of the website hosting the content, or a third party service. The underlying programming may be in any programming language outside of the HTML framework for the pages.

Other websites may contain content with effects or built-in functions, such as a link protected by a scripted button. Scripted functions are often created in JavaScript; e.g. the Toolbar at the top of pages shown in The Internet Archive shown and discussed in 8.1.4 Examples from The Internet Archive, p. 140-142.

Scripted content works as programs running either server-side, or in the case of JavaScripts in the browser. As such, scripted content is not a direct part of the website or web page, and contains code and functions that cannot be stored or copied by the web harvester.

Streaming Content
Streaming content, e.g. a live transmission of an event, is delivered in data blocks that are rendered into content in real-time at the user’s end. It is not delivered in the form of files and cannot be stored or captured as such.

Embeds of Some Types of Content
Embedding something on a web page means calling in content from outside of the primary domain where the web page itself resides, and placing it so it is represented on the page when shown in a browser.

Embeds can be commonplace content that can be easily copied (e.g. images), but they can also be for content that cannot be stored, either because of its nature (e.g. interactive elements that run as programs on a server), or the nature of the service (e.g. embeds from social

services that allow showing their content, but protect it against being copied).

Thus, while embeds rely on HTML code and as such are a Web technology (see 4.1.3 HTML: HyperText Markup Language), the embedded content may or may not be possible to harvest.

To users in general, the most commonplace types of embeds may be changing advertisements such as banners, or videos from external services such as YouTube.

These two types of embeds, along with embeds of social media streams (e.g. feeds for topics related to that on a web page) or discussions (e.g. discussion threads for articles in news media, hosted and run by an external social software service) – are probably the embeds where users will most likely be aware that this web element is not a integrated part of the web page they are looking at, but content coming from “somewhere else”.

There are however, many types of embeds. Embeds can also be images hosted at an external service, applications that cannot run directly on the web page (e.g. simulations), specialised information services such as live weather forecasts, etc.

The shared trait of all embeds is that they are calling content which is not part of the web page itself, and that the content placed on the web page is not hosted (stored) on the primary domain of the web page itself. The content may be running on the same server and placed somewhere under the same domain, but it will be outside of the main webpage structure.

For web archiving purposes embeds represent two main challenges. The first is the question of including the content if possible. The second is that of whether it is possible or not.

As a general rule, including embeds in a harvest process, so that their content is stored with an archived copy of a web page and shown

correctly when viewing said copy – may be done for images in most cases and for videos in some cases, but often an embed depends on an external service which cannot be saved in a fully satisfactory manner – it is simply not technically possible.

This has to do with how the embed is handled on the web page, and with how the content is created or stored at the provider that it is fetched from.

Examples of embedded types that cannot be archived are:

- Interactive content running on an external computer, using other programming code than HTML,
- Streaming content, e.g. YouTube videos, or live broadcasts,
- Content from social media, which is controlled and restricted by their APIs (see also 6 Social Media).

In the case of images an embed may be simple: In the HTML code a command may say “place an image here (along with specifications on size, framing, etc.) and fetch it from this URL”. The URL may then direct to a storage on the web page’s own domain, or it may be from an external domain. In the latter case, the image is still represented as a file on the web page via the embed.

Here the challenge is simple: In order to get an image file included and placed in an archived web page, the image in question must be stored directly along with the web page. This will be successful if the saving method allows for fetching the external content (see “Delimitations on Width and Depth” below).

But when embeds are calling content that cannot be harvested, then naturally it will not be. This is a challenge to the crawler in the sense that it is something that the crawler simply cannot do.

What the crawler can and will normally do, however, is to copy the embed code since this is a part of the HTML code in the HTML file for a given page. This can cause a side effect of “online leaking” when viewing or evaluating the harvested content. Please see 7.5 The Risk of Online Leaking for more on this topic.

Hidden pages

Any website domain can contain pages that are not visible or accessible from links on the website itself, e.g. drafts of content, content meant for later publication or taken off the menu after its relevance has expired, or content meant for sharing with specific contacts via email.

If the content is neither in the menu or linked to elsewhere in the pages, the web harvester will not encounter a link for it, and it will therefore not be harvested.

Delimitations on Width and Depth

Websites may have pages that are located very deeply in the menu levels, and possibly even deeper if located under pages that are already located deeply in the website structure. If a web harvester is delimited to, e.g. crawl 25 levels on websites, then levels under the 25th will not be retrieved.

Similarly, if content displayed on web pages is too far removed from the web page’s own domain in respect of the delimitations set for the harvest job, then said content will not be copied. If for example, a harvest job is allowed to go two steps away from the targeted websites in the seed list, then it may fetch images that are hosted at another domain (step 1) and stored in a separate “storage domain” (step 2). But if the location of the image is actually three, four or more domains away, then the two step delimitation will not allow the harvester to retrieve the image.

Older websites and web pages found in archives are more likely to show such adverse effects due to delimitations that were necessary at a time when internet bandwidths were smaller and storage space more

limited, than newer pages that were harvested with better technological capacities allowing for broader delimitations.

But all harvesting jobs are meant to acquire specific content and end after a reasonable time and will therefore have delimitations by necessity – with the potential side effect of some content being too remote in either width (how far removed from the targeted websites it is), or depth (how deeply it is placed in a website structure).

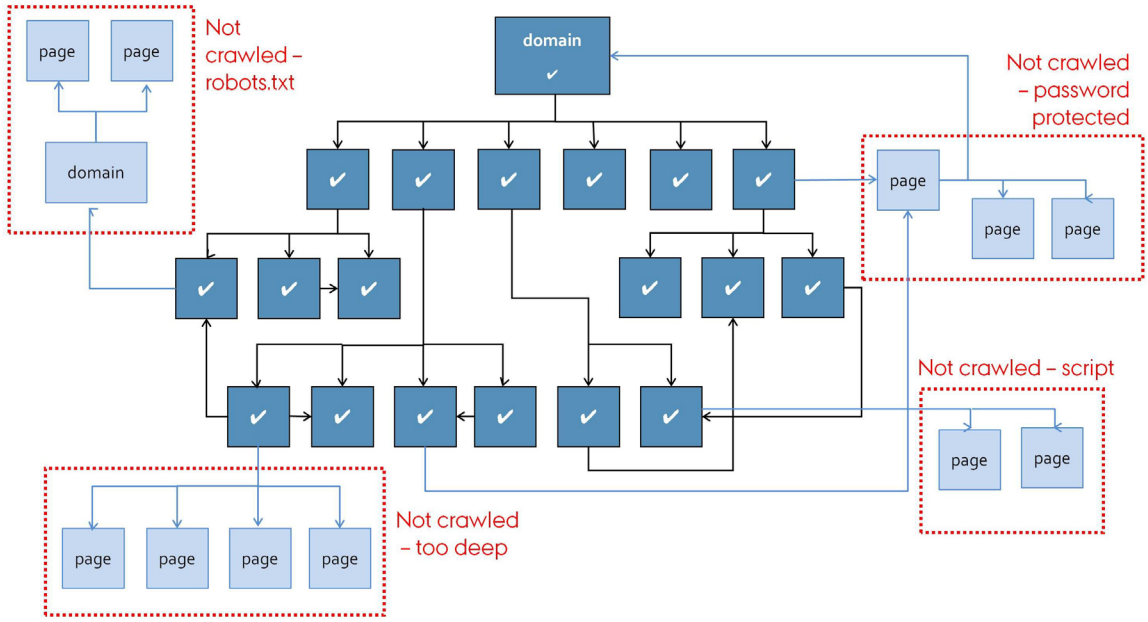


Figure 18: Examples of content not saved in a harvest job.

5.2 URLs Are Changed

In order to preserve a connected whole of the pages that make up websites, and of the relationships between different websites, the harvester has to change the URLs in the harvested content.

In the institutional archives, a page is saved with an archive address, a timestamp, and the original web address that the content was harvested from.

The timestamp for a page, or for a page element such as a picture, will reflect the exact time of harvesting that specific content.

URLs found on a page are changed in the same manner; they are also given an archive address, and the same timestamp as the one given to the harvested page. This does not mean that the other pages or content types that a page links to were harvested at the exact same time as that page.

When reconstructed – when viewed in the institutional archive – the program that reassembles web pages and websites into a resemblance of the online content (e.g., the WayBack Machine, see 8.1 The Internet Archive) will not have copies of the linked content that exactly matches the timestamp on a given page. Since harvesting takes time, other pages or content will have been harvested earlier or later. When the user tries to pursue a link on the starting page, the WayBack Machine will use the timestamp in the archive link as a reference and retrieve content that is as close as possible to the timestamp from the starting page.

This will normally lead to time jumps as described in 7.6.1 Time Jumps in the Content and shown in 8.1.4 Examples from The Internet Archive, p. 143-146, which is something that one should be aware and wary of. In other cases, the new content that the user is attempting to access may not have been archived, and the user will reach a message accordingly. So the changed URLs found on any archived page in an institutional archive, will reflect the timestamp of that page, not as exact addresses to existing content, but as an internal technical way of attempting to locate the content, *if* it is also available in the archive.

In small-scale archiving where content is preserved in a local storage, the URLs will be changed to reflect a folder structure[8], as described in 9.4 Archiving Websites or Specific Content from Websites.

The changes in the URLs are necessary in order to maintain coherence in the archived version.

[8] Web harvesters for local archiving which also include timestamps for the harvested elements may already exist, or emerge. But for coherence in the elements in a local storage, the internal URL structure in the saved content will have to be based on an internal “archive version” structure.

If relevant, e.g. in order to determine whether a web page or element still exists in the live Web, or how it may have changed, the original web address for the live Web can be retrieved by deleting the prefixes that connect the saved content in the archive. This will not always give the full URL with a HTTP or HTTPS protocol call, or the www. prefix if relevant, but the original web address will work in a browser if the web page still exists online. See also 8.1.3 Internet Archive URLs for a breakdown of the elements in archive URLs, and 9.3 Copying URLs, p. 198-199 for an explanation of browser's handling of web addresses in browsers.

Examples of the changed URLs in locally archived websites are shown and discussed in 9.4.3 Checking the Quality of a Harvest, p. 209-215.

6 Social Media

Takeaways

♦ Social media are technically difficult to harvest because their content is controlled, handled, and delivered by the respective services' Application Programming Interfaces (APIs).

♦ Most social media services restrict access to their data to extents where only limited or no content can be copied systematically.

♦ Systematic and automated harvesting of a social medium requires access to the API for the medium in question, wherefore the specific conditions set up by a service provider defines what can or may be done.

Social media, often abbreviated as SoMe, is a term for the overall palette of various platforms where people interact by creating profiles (also termed as user accounts) and posting content for others, privately or publically.

Social media are problematic in several ways when it comes to research. There are four issues that cause social media to stand apart from traditional web content:

♦ Social media change much more rapidly than websites or web pages in respect of content, as millions of users contribute and make changes on a daily basis,

♦ The underlying technologies (APIs) and their content cannot be harvested as HTML-coding and files,

♦ APIs frequently change whereby working data retrieval methods, when or if they exist, may stop working and call for new ones,

♦ Most social media restrict harvesting as well as API access that would make data harvesting possible, even for researchers.

Changes to the APIs for various social media occur, and when the underlying software changes, so does the means and possibilities for accessing the content, and especially for harvesting it.

Moreover, the service providers' policies, conditions and possibilities for access at levels that allow for harvesting social media content has notoriously varied over time, often with sudden changes where open and free researcher access with short notice became very restricted, or expensive, or closed down entirely.

It is therefore not feasible to give a general description of how social media may be archived, when and if possible.

But social media have a strong presence and impact on the Web. Companies, news media, politicians, organisations, private persons,

and interest groups are all likely and more or less expected to have a visible presence on them. On a daily basis, millions of users post, comment, and share anything from jokes and daily life updates to opinions and news, fake or real.

Rumours, protests, movements of any kind, conspiracy theories, trends, etc. may start or spread or thrive on social media, and political campaigns may be won or lost depending on how they act and are reacted to in the social media landscape.

So, from an archiving perspective that wants to preserve cultural heritage, as well as from a research perspective that wants to be able to track and analyse cultural or societal developments – it is deeply problematic that social media with large amounts of crucial content, defy systematic and stable harvesting and data access.

The social media services are all controlled by privately owned companies, and their policies for access are decided and controlled as such. Some variants of social media may allow some types of access for preservation or research, but this landscape changes so frequently and dramatically that only a general presentation can be offered.

How and why privately owned platforms came to be perceived and effectively used as primary spaces for public debate is a question of great philosophical potential, but here it must suffice to attempt to give an overall explanation of the resulting circumstances.

6.1 APIs and API Access

The social media differ from traditional Web content. Although they have domains and portals that can be accessed in a browser, they rely on their own technological systems: APIs which stands for Application Programming Interfaces.

The APIs handle user accounts, content, and how the content will be represented when displayed in different contexts, e.g. if accessed in a browser, or viewed in an app for a mobile device, or represented as a

news feed via an embed on a web page (see also 5.1 Challenges for Web Crawlers, p. 79), or in debate sections for articles where a social medium is used as a third party provider for engaging the readers.

Social media posts are stored in large databases which are handled by the API belonging to the service in question. The databases may contain large amounts of unique data cells for each post, with information such as; Post ID number, User ID, Username, User profile name, Date, Geolocation, Text, Image locations, Video locations, Web references, Likes, In response to, Number of responses, Number of shares, etc.

For example, a user may or may not have included a web URL in a post. This will be handled as data cells with URLs from the main text. An image included may come from a wide array of places, and the data cell for images will then contain the image location (usually a URL). Geolocation may or may not be provided, etc.

When accessed from a browser, in an app, as a feed on a web page, or other, the API handles how, and which parts of, the data is set up and shown.

If harvested, the data is delivered in the format of unique cells that may be imported to or read as a spreadsheet format.

The visual expression of the original post will be lost when harvested as data, since the intended placement and selection of elements are handled by the API, depending on the context where the post will be shown.

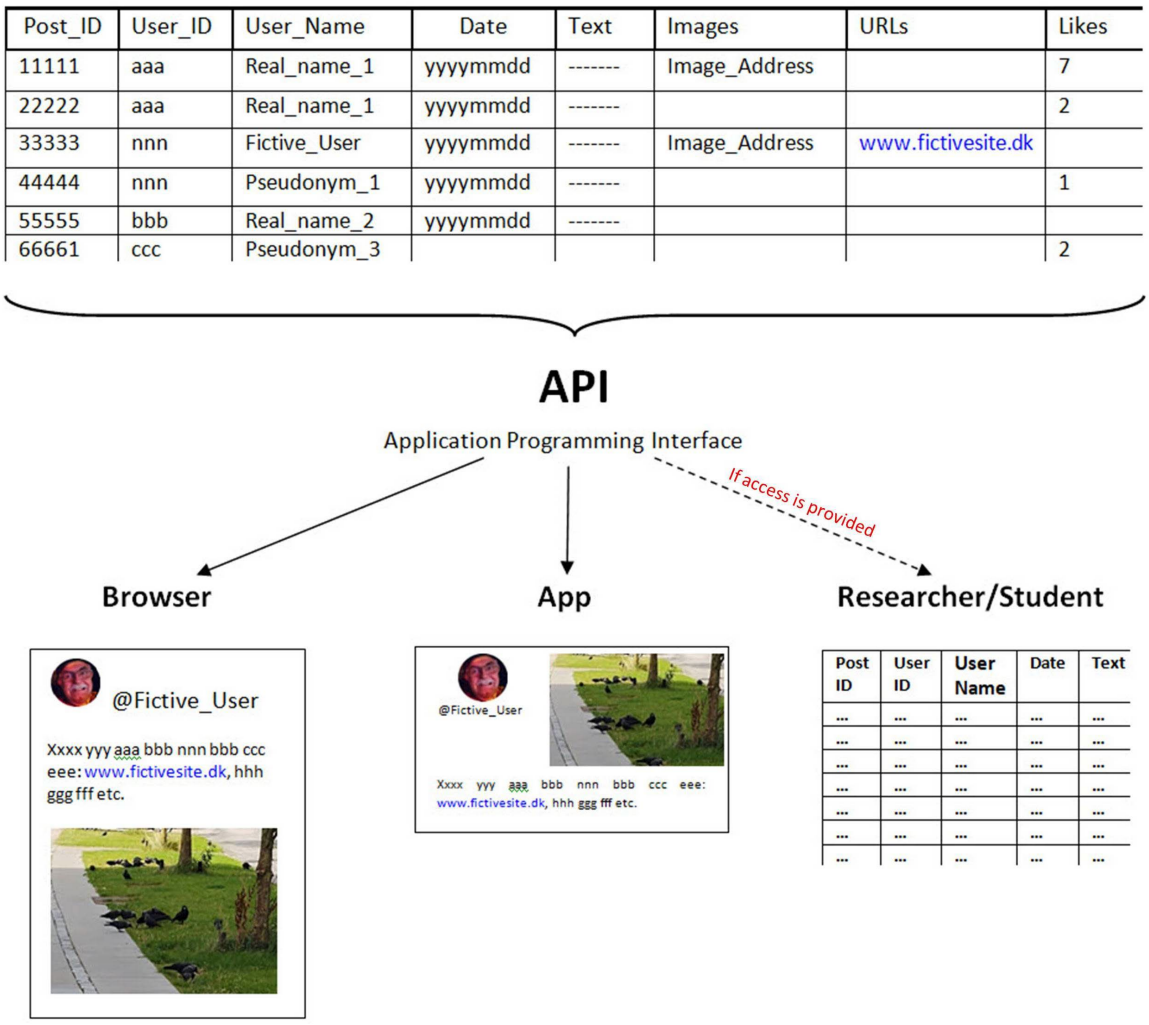


Figure 19: Social media posts are handled by the API depending on the reception format.

For institutional archives the challenge of archiving and preserving social media in a recognisable way has sometimes been resolved, at least in timeframes where harvesting was at all possible. There have been collaborations between some social media services and institutional archives, e.g. between The Library of Congress and Twitter from 2010 to around 2017, where the collaboration decreased. (Vlassenroot et. al. 2021, p. 109).

Thus, for some topics, some social media, in some time intervals, one may be able to find archived content that reflects the original content as it would have looked when visited online in a browser. But in general, the institutional archives face severe problems in preserving social media as cultural heritage.

As an example the WARCnet researcher network[9] published a series of interviews with web archivists who had been involved in special collections related to the Covid-19 pandemic.[10] The interviews were conducted in 2020-2023. They include discussions of what was harvested, and social media are consistently mentioned as problematic.

For this reason archives chose to focus mostly on websites, e.g. Ben Els: "We tried to concentrate on websites, because social media platforms raise a lot of technical difficulties and they are also more expensive to harvest in term of data budget, with often unreliable results. So we have included some Facebook pages for example, but we haven't saved a lot of social media data; we have archived much more news media, websites and Twitter, which can be archived much more effectively than other social media (Shafer & Els 2020, p. 4)."

It should be noted that at the time of the Covid-19 pandemic from 2020-2023, whereafter the virus is still global but no longer considered "a public health emergency of international concern" by WHO[11], Facebook allowed harvesting of "public" pages; that is pages where the content is open to anyone logged in to Facebook, e.g. organisations or politicians, and Twitter allowed for large scale harvesting for research purposes. Facebook is still open to harvesting of "public" pages as of 2024, while Twitter changed ownership in 2023, was rebranded to "X", and free developer and researcher access was closed. Paid solutions for access to the X API exist, priced in 2024 at $100 per month for access to 10,000 posts per month, and $5,000 per month for access up to 1,000,000 posts per month.[12]

---

[9] https://web.archive.org/web/20230901181014/https://cc.au.dk/en/warcnet/about
[10] https://web.archive.org/web/20230902201447/https://cc.au.dk/en/warcnet/warcnet-papers-and-special-reports
[11] https://web.archive.org/web/20240820140453/https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-%282005%29-emergency-committee-regarding-the-coronavirus-disease-%28covid-19%29-pandemic
[12] https://web.archive.org/web/20240820141159/https://developer.x.com/en/products/x-api

According to the different interviews conducted and published by WARCnet, most of the archives did harvest some social media content, but their capacities and priorities differed. Most of the interviews have direct mentions of difficulties with social media.

The status of social media access and harvesting for archiving and research purposes has been changing often and dramatically, both technically and in respect of the levels of access allowed by the companies. In many cases some content can be harvested in some ways, but the technical difficulties and “often unreliable results” mentioned by Ben Els above are sadly the status quo that one must expect, both if searching for specific content in institutional archives, or if attempting to harvest data for a project.

7 The Characteristics of the Archived Web

Takeaways

♦ Archived web is an extremely useful scientific resource with vast potential; but as with all cultural artefacts it comes with limitations. Those limitations can be negligible in some cases, while it may be necessary to find ways to handle them in others.

♦ Some content cannot be preserved, and will in that case normally be lost. This does not affect all archived web pages, but will cause occasional gaps.

♦ It can be difficult or impossible to see if content is missing, but the metadata for archived copies, or studying their source code (HTML) can often help determine what was archived, and what was not. Institutional web archives have provided better access to metadata in later years, making it easier to study the provenance and see exactly what an archived copy consists of.

♦ Various types of temporal inconsistencies can occur, for example as changing dates for various copied pages when attempting to inspect an archived website.

♦ If an archive is not isolated from the internet, web content can sometimes be “drawn in” by an embed code and make it appear as if the archived copy is more complete than it actually is (“online leaking”).

This chapter will introduce how the differences between archived web pages and their online originals may be encountered by the users.

This is important to know in order to avoid mistakes, but the many details that one may have to take into account and the explanations of their nature and consequences may have an adverse effect of causing more worries than necessary.

In most cases the finer details will not matter much to users of web archives. Temporal inconsistencies may be interesting or even mildly funny paradoxes of no importance to the quality or usefulness of the materials of interest. Missing content may not affect the archived pages visited at all, and if it does, the user may have no other reaction than, “Oh, well, I couldn’t find that”.

It is simply that, users of archived web materials need to understand that the archived content is not flawless, and therefore also need to know the characteristics of the possible pitfalls; what they are are, how to spot them, and how to handle them.

But before we go into the gaps and errors that may be encountered, let us start with the *most important characteristic* of archived web:

Archived web is an immensely valuable resource, both as cultural heritage, and as research material.

While it consists of artefacts that are not always entirely or flawlessly complete, it offers preservations of significant content in numbers that are no less than extreme. The amounts of data preserved in the form of archived web easily surpass the amount and quality of artefacts left from earlier cultures or historical epochs.

Never before have the actions, ongoings, sentiments, trends, decision processes – in politics, business and industry, institutions, news media, entertainment providers, interest groups, and individuals, been documented or preserved at the level of detail offered by ongoing and

systematic preservation of the Web, with content from all levels of society.

Cultural artefacts are often flawed or incomplete, and no cultural artefacts exist in their original context. But they are what we have. In archived web we have more cultural heritage preserved than earlier cultures or scholars have been used to. We just have to understand that it consists of artefacts, and that they, like all other artefacts, come with limitations.

The driving purpose of archiving websites is of course the preservation and the possibility of revisiting the content. But as mentioned in 2.1 Digitised, Born Digital, or Reborn Digital and further explained in explained in 5 The Web Archiving Process, the archived content is changed from the originals, e.g. in the sense that it no longer exists in the context of the live web alongside with the pages it may link to, that some content may be missing, that URLs are changed to archival versions, etc.

Since web archiving, especially on the large scale of institutional archives, may attempt to avoid having overly many copies of specific files, and therefore omit repeated copying of files that are already in the archive, another and more subtle characteristic of the archived web is, that it is pieced together from fragments stored in different places in the archive. While the visual impression of an archived page may look very loyal to the online original, this is a difference that can have subtle implications.

For example, a copy of a web page with a YouTube movie embedded may have a preview still of the movie captured successfully, and this web element may thus appear exactly as it did on the live Web original. The difference however, is that clicking on the video in order to start it may not work, or may instead call the video from a live YouTube resource. This specific type of problem is addressed in more detail in 7.5 The Risk of Online Leaking, and 7.6.3 Checking Against Online Leaking, Local Archives, and 7.6.4 Checking Against Online Leaking, Institutional Archives.

Some effects, or possibilities of interaction that were on the live page may be missing, and it will not always be possible to see or determine if this is the case.

Questions of provenance; where does this come from, how is it pieced together, are there signs that something might be missing – have been a bigger issue in the early days of web archiving than it is today (2024). The web archives have been working towards support of better access to such metadata. Examples are given in 8.1.4 Examples from The Internet Archive, and in 8.2.5 Examples from Netarkivet.

It is worth mentioning that the older the content one attempts to study, the more erratic the quality usually becomes, both of the content itself which is more likely to show signs of gaps and in the level of information and documentation in the metadata. When the phenomenon of archiving web content was new the technologies were less developed, and a full awareness that documentation of how the content had been archived might be important had not developed.

As mentioned before, all such observations may lead to unnecessary concerns or reservations about the usefulness of archived web. But in most cases such concerns are of little practical relevance. Let us therefore sum up the pros and cons:

7.1 Pros and Cons of the Archived Web

The “pros” of archived web are significant, and most readers should be able to take comfort in them.

While there are “cons”, they can be handled with proper awareness and some are only relevant in special cases, e.g. if the exact look and feel of the original web page at the time it existed is a primary object of study.
The “cons” primarily stem from the way that the archiving process changes the content, and the limitations that prevent some parts of the content from being saved, as explained in chapter 5 The Web Archiving Process. An additional discussion of the implication of the

changes to URLs and the resulting temporal inconsistencies (time jumps), and a problem that can occur with HTML code calling online content from the live web will be addressed further on in this chapter.

Pros

- The textual content is preserved. An archived web page consists primarily of the HTML file, in which the main text for a page is normally contained.

- Images are the most primary content type besides from text, and images for web pages are normally harvested and inserted as stated by the HTML code in the primary file.

- The overall layout and impression of the content of an archived page is therefore a loyal approximation to the main content as it would have been encountered on the live Web.

- The large amounts of data found in archives from systematic and ongoing harvesting make complex studies possible, e.g. where specific terms were mentioned at a given time, or how or where a topic has been treated historically, etc.

Cons

- The content found in web archives is not a 1:1 copy of the original content. It may have gaps in the form of missing content, it is taken out of the original online context, there are changes due to the technological process of handling the archived copied (see 5.2 URLs Are Changed), and decisions and settings for the harvesting process that retrieved the content are also decisive for the levels of detail that may or may not be included.

- It can be difficult or impossible to determine whether some content is missing.
- There may be temporal inconsistencies to take into account.

- Online leaking may give a false impression of specific elements of content having been archived while they actually have not.

- Large amounts of similar copies of the same content can make it difficult to determine which version or versions one should choose for a study. A set of criteria may be needed for a consistent choice of data, e.g. “the first ones” or “the middle ones” from a list of copies of a number of selected pages.

In summary, the content found in web archives may be compared to other types of cultural artefacts that may not be fully complete, but still serve as evidence of “what has been” and “what has occurred”.

The trustworthiness, especially of the textual content, as well as the amounts of data collected, make web archives very solid as sources and resources for studying phenomena from the 1990’s and forward.

7.2 Missing Content

Missing content can be encountered in four ways:

- An entire page is missing,
- One or more elements are clearly and obviously missing,
- One or more elements are missing, but it is not visible that they should have been on the page, or what would have been the nature of the content,
- One or more elements seem to be present, but are in fact not archived.

The first three instances have in common that there is content that is not shown, and they also share the same possibilities for being resolved, namely to look further in hope of finding the missing content or something that might clarify what it could or should have been. Of course, looking for missing content is hardly relevant if one already has a satisfying artefact, and e.g. only needs the text.

The last instance occurs as a phenomenon that has already been touched upon and referred to as “online leaking” in the beginning of this chapter, and under “cons” in 7.1 Pros and Cons of the Archived Web.

Attempts of finding, identifying, or verifying missing content are called “data mining”, that is “digging deeper or digging around for something that one hopes to find, based on one’s needs and the clues that one might have”.

7.3 Missing Pages

Missing pages may be encountered in different ways depending on whether a web archive is local or institutional.

For local archives, please refer to chapter 9.4.3 Checking the Quality of a Harvest where inspection of the local addresses are the primary key to determining whether specific content has been preserved, and 9.4.4 Attempting Repairs for what may be done if it hasn’t.

If an institutional web archive reports that a web page is not in the archive, then the address for the page has either never been harvested or attempts to do so have failed.

The web archive may report that the page is online and offer to attempt to save it. This may be tried with the understanding that, a) it may fail if the page resists archiving, and b) that the page, if successfully copied, will be an entirely new copy that does not reflect possible earlier versions of the page.

Failing to find a page in an institutional archive, it can be searched for in other archives where a copy may exist.

If all attempts to find a web page in an institutional archive fail, the last option is to see if the page is online, and in that case copy it directly. If it is a page that resists being copied, alternative solutions such as screenshots may be applied. See 9.2 Basic Archiving, Single pages.

7.4 Missing Content Elements

When or if specific content was not successfully archived, this may either be obvious from missing images or content icons, or a message from the archive stating that some content is missing here – or is may be less obvious if no message of sign of missing content is shown.

If an archived web page has a blank space that seems large or out of place, this is a sign that some content would probably have been there on the original, live page. As an example, figure 20 shows a version of the front page for myspace.com from The Internet Archive on June 14, 2006.

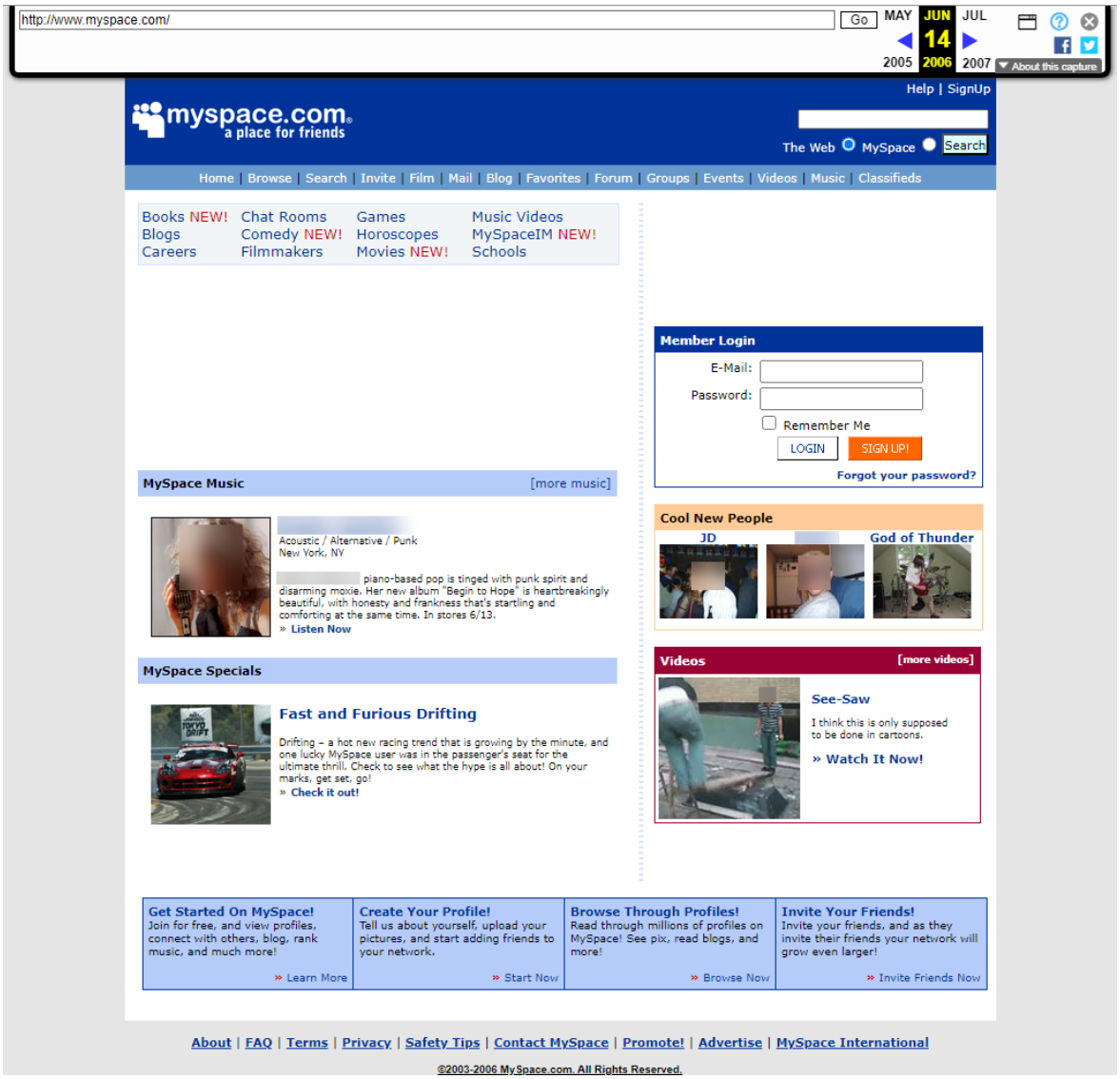


Figure 20: MySpace front page at The Internet Archive, from June 14, 2006. Details blurred by the author.

Both columns on the page have blank spaces at the top, suggesting that there should have been some kind of content. If the rest of the content is sufficient, e.g. if one is only looking for the main text on a page, then the missing elements may not be a concern. But if it is important to get a fuller impression of the page and determine, e.g. if there could have been a video, or an important illustration, or an interactive script that carried important meaning, then the archived page is an incomplete artefact.

The missing content can possibly be retrieved, found, or at least identified either from an alternative copy or with data mining. An example is given in 8.1.4 Examples from The Internet Archive, p. 141-143 where a video is not archived, but is traceable via provenance data supplied by the archive. As long as the original content exists online, it may be copied by various alternative means from its source, to a local and supplementary data collection.

If the content cannot be found then there is nothing to do about it but to either discard the artefact, or accept it as it is.

The first and easiest way to check for missing content is to look for other copies of the same page from dates as close to the incomplete copy as possible in the archive one is looking at, and possibly also in other archives (see 8.4 Other Web Archives).

This may result in finding a more complete copy with the content included, or a screenshot that suggests the nature of the content. Such alternative copies may be better than the one first found in respect of being suitable for one's research – or they may be relevant as supplementary resources.

There is a risk that must be considered here, namely that the content found on the more complete copy may not be exactly that which was missing from the page that one started from. Another archived copy of a web page will at best be a version that was archived at a time close to the incomplete copy, and the original content may have changed on the live page in the time interval between the two archived versions.

Another option for handling missing content or the suspicion of it will be taking a look at the HTML code as described in 4.1.3 HTML: HyperText Markup Language. By doing so one may be able to find an HTML code that specifies an embed or other clues to the nature of what should have been present at a specific part of the page. But this depends on the exact way that an embed or script was added to the web page, and whether the method for harvesting the copy in question was capable of capturing HTML traces of it.

7.5 The Risk of Online Leaking

If a page on the live web had embeds calling in services or content that cannot be harvested in a web crawl (see Challenges for Web Crawlers, p. 79) – then the archived copy of that page will normally have the embed code, but not the content.

This can result in an archived copy of a web page showing content that is actually coming from the live Web, and this can lead to two types of mistakes:
1) By giving the impression that the content showing on the archived page is preserved in the archive, when it is in fact something that is not archived but comes from an online resource that happens to still exist outside of the archive.

Or,

2) By adding present-day content to a preserved web page copy that was created earlier, resulting in a temporal inconsistency (see 7.6.1 Time Jumps in the Content, p. 107).

Advice on trying to determine whether online leaking is taking place in institutional archives or local archives respectively is provided in 7.6.3 Checking Against Online Leaking, Local Archives, and 7.6.4 Checking Against Online Leaking, Institutional Archives.

7.6 Three Types of Temporal Inconsistencies

When looking at archived web content one is likely to experience a few types of temporal inconsistencies, where some content may appear more or less anachronistic or paradoxical.

In most cases this will not affect the overall experience or relevance of the content, and in many cases it may be disregarded.

However, it stands to reason that discrepancies that go unnoticed may lead to misunderstandings, wherefore it is important to be observant and aware of what may occur.

There are three types of temporal inconsistencies:

Time jumps in the content, changes captured during the process of archiving, and online leaking.

7.6.1 Time Jumps in the Content

When looking at an archived web page, one may experience three different types of time jumps in the content. They are closely related, and stem from the ongoing process of archiving:

1) The elements of the web page may not have been archived at the same time.

In a local archive where one has created a copy of a website on a specific date, this is a minor problem, but there still may be hours between the time that a page was harvested, and that the images on it were. For example, the web crawler may have harvested the page relatively early in the process, but the images – possibly hosted on an external page or a page far removed from the one where they are meant to be shown – may come from a URL that the harvester registers, but does not reach until later in the process.

The level of possible discrepancies between time of harvesting single web elements rises dramatically in institutional archives. In the

ongoing process of harvesting and re-harvesting web pages, the large web archives will often skip files that are already in the archive.

For example; an image, an embed, or a CSS style sheet[13] for a website may have been harvested in a previous run of the harvesting program and, in order to save space and time, omitted in a new harvest.

As a result, the archived web page one is studying, with a time stamp for a specific date, will often consist of content that is pieced together from earlier – and sometimes later[14] – harvests. The web archive will display the best possible reconstruction of the page as it looked at the date of the time stamp, but the content shown may be pieced together from a number of actual harvests.

If metadata is accessible as in the examples of The Internet Archive and Netarkivet (see 8.1.4 Examples from The Internet Archive, and 8.2.5 Examples from Netarkivet), then it is possible to examine the page resources and their harvest dates. While this is not relevant for a basic look at "how the web page looked at the date for the primary page capture" it may be interesting, e.g. if one wants to follow up with finding the earliest occurrence of an archived image, possibly after having found a later copy of the page where the text may provide new or different information about it.

In all cases one must expect that archived web pages consist of elements that were not harvested at the exact same time.

2) Some elements on web pages may not have been archived, because the content is embedded content of a type that could not be preserved, but which is drawn in and shown on the archived page from the live Web by the HTML embed code. See also 5.1 Challenges for Web Crawlers, p. 79, and 7.5 The Risk of Online Leaking.

---

[13] CSS means Cascading Style Sheet. It is a background HTML coding that defines effects, layout and typography across a number of pages.

[14] E.g. if capture of specific content failed in a number of harvests, and then succeeded, one may find that an element on a page was harvested later than the page itself.

Such a mix of archived content from the past and present-day content from the live Web is of course a temporal inconsistency. The online content can for example be ads, weather forecasts, or feeds from the latest comments on a topic on a social medium.

As long as the focus remains on the main text of a page and other primary content such as images while other elements are disregarded, this will not be a problem. But there are risks of mistakes if the live content is taken as relevant for the preserved content.

For example, a sunny weather forecast at an article that expresses “sunny and happy feelings” may be perceived as connected, while the article may in fact have been written and published under dark and gloomy weather conditions. Or if ads for raincoats are shown in the “sunny and happy” article, the article’s notions may be perceived as ironic.

The primary advice if a page is viewed as a whole and interpreted as such, is to be wary of content that does not clearly belong to the page itself. See also 7.6.3 Checking Against Online Leaking, Local Archives, and 7.6.4 Checking Against Online Leaking, Institutional Archives.

3) The third type of time jumps are not readily at hand when looking at a specific copy of a web page, but are very likely to occur if or when one decides to follow a link, e.g. in a website menu shown on the page.

More often than not, the new page that opens will have a different time stamp from the one it was accessed from.

Again, this is due to the ongoing archiving process, but one cannot be positively sure that “the next page” was omitted from harvesting at the timestamp date for the previous page due to an identical copy already existing in the archive. This may well be the cause, but there may be other reasons; for example the page visited from the previous one may not have been harvested on the same date for purely technical reasons. The only thing that is certain is that the new page is the

version that is closest to the first one in the archive. So there is a risk that the new archived page may have different content than it would if it had been harvested on the same date as the previous one.

This means that not only web pages, but also websites appear as reconstructions; the best possible approximations to the live content they were copied from.

It is important to be aware that entering a copy of a web page for a specific date does not imply that the rest of the website content can be studied as a whole from the same time as the point of entry, and to observe caution by taking such discrepancies into account. It cannot be assumed that the result from a search for a website on a specific date will accurately reflect the original website as it appeared online at that date.

As one studies an archived website the time jumps may lead further and further away from the starting point, the date and timestamp of the copy first entered. If one is removed from the starting point by following a link on an archived page to another page, and then follows another link on the new archived page, the next result may be closer to the starting point – or further away from it.

An example of this type of time jumps is given in 8.1.4 Examples from The Internet Archive, p. 143-146.

7.6.2 Changes that Occurred During Archiving

An archiving process takes time, and web pages and websites change often and regularly. On any website there is a risk that changes coincide with a harvest process.

The more often the content on a website changes, and the longer the archiving process takes, the higher the risk becomes of discrepancies between the various pages archived.

Niels Brügger has reported this example from 2000:

“During the Olympics in Sydney in 2000, I wanted to save the website of the Danish newspaper JyllandsPosten. I began at the first level, the front page, on which I could read that the Danish badminton player Camilla Martin would play in the finals half an hour later. My computer took about an hour to save this first level, after which time I wanted to download the second level, “Olympics 2000”. But on the front page of this section, I could already read the result of the badminton finals (she lost) (Brügger 2005, p. 22).”

Although the example is from 2000, rising speeds in bandwidth, programs, and computers have not eliminated the problem. Any harvest job takes time, and changes in the content may still occur while the harvest job is running.

A mismatch between two articles or pages where an update to the last one captured contradicts the first one captured is one example of what may occur. Another possibility is that the harvest is allowed to go on, until there are no pages that differ from those already captured. With such a delimitation (see more on delimitations in 5.1 Challenges for Web Crawlers, p. 82-83), the changed front page would also be harvested. This would result in more copies of some pages, reflecting the changes that occurred – in different versions of the same pages from the same harvest.

Institutional archives attempt to solve this with a strategy of harvesting websites that change fast and frequently, such as news media sites, several times daily.

The problem is related to, and may add to the previously described question of time jumps when going from page to page in archives versions of websites (see 7.6.1 Time Jumps in the Content, p. 107-108).

7.6.3 Checking Against Online Leaking, Local Archives

More often than not online leaking will not be a problem, since the main content of text and images on web pages can normally be preserved to a satisfactory extent, and will likely be the content of most interest.

However, if one needs to be sure that the content viewed is exclusively the same as the content actually stored, there are ways to do so if websites or pages are stored locally. Please see 9.4.3 Checking the Quality of a Harvest for explanations on localised URLs.

In that case, the following methods may be helpful:

1) First of all, observe the URLs shown in the address line of the browser, and on mouseover (usually shown as a small popup in the bottom left corner). If an address is shown as localised on your hard drive, e.g. starting like “file:///C:/My Web Sites/”, then the main content for a page is at least stored locally.

This does not rule out online leaking, but it can help against making the mistake of following a URL which was not stored and mistaking it for stored content. If the URL shown is a regular web URL (starting with HTTP, HTTPS, WWW or similar), then one is looking at something which is online instead of stored in a local archive.

2) Examine the archived content in a reference browser without internet access, and where the cache has been cleared.

The reason why the reference browser must have its cache cleared is that some online content may be cached (temporarily stored in the computer’s local memory) in order to show web content faster and more effectively. Cached content may still be shown, even if the browser’s internet access has been closed, and thus still appear as if it were present in a local archive.

To clear the internet cache, one must use the relevant option in the selected web browser. Since this differs slightly between browsers,

one should do an online search for “clear the internet cache + name of the browser one is currently using”; e.g. “clear the internet cache Mozilla Firefox”.

Please be warned that clearing the cache fully may also delete access codes and usernames that the browser was set to remember. This will happen if one does not opt out of clearing cookies or full history. A loss of stored access codes can be inconvenient for the user, wherefore it is recommended to make sure that all such information is backed up elsewhere. One can do a search for “back up stored passwords and usernames + name of the browser one is currently using” and create a backup as instructed before proceeding with clearing the cache.

If one has an extra browser where it is not important to keep the web history, passwords etc. this will help make the verification procedure somewhat easier, since precaution and backup can be omitted.

The full procedure in any case is:

Clear the cache in the test browser. Close any internet connection for the computer, or block the test browser in the computer’s firewall. Verify that the test browser has no internet access, e.g. by attempting to start an online search. This should fail due to the lack of an internet connection.

The archived content can now be inspected in its pure form any without risk of online leaking, by opening it in the test browser: Right-click on an archived page (ctrl+click in Mac OS) and select “open with [name of test browser]”.

Recipes for temporarily disconnecting the computer from the Internet, if this is preferred to temporarily blocking the test browser in the firewall, can be found by searching online for “deactivate local network devices”.

With a cleared browser cache and no internet access, the archived copy will show only the content that exists in the archive.

7.6.4 Checking Against Online Leaking, Institutional Archives

For institutional web archives, the chances of ruling out online interferences are more complex than in local archives. Basically one cannot prevent online content from being shown when looking at a page in an archive.

A primary question is therefore whether the archive one is looking at can at all connect to online content. In the case of the Danish “Netarkivet” the content is accessed via a protected Citrix solution that provides a remote workstation for Netarkivet’s interface. In this case nothing from the live Web can come between the user and the content that is being viewed, and thus no risk of online leaking is present. The user will be looking at archived content and nothing else.

But if the web archive you are looking at is not protected against online bleeding, or if you do not know, then a bit of foreknowledge may help:

Videos are often embeds, as are ad banners, active weather forecasts, active news feeds from external websites, or feeds from social media.

Such content may or may not have been archived. The HTML code and/or the provenance metadata are the only ways to positively ascertain whether specific web elements are archived content, or shown online from the live web. An example of using the metadata for this is given in 8.1.4 Examples from The Internet Archive, p. 141-143.

For pages viewed in archives with open public access, online leaking is a risk. In such cases one may be helped somewhat by using mouseover on the content: If a URL is shown and it is not an archive address (e.g. starting with “https://web.archive.org/web/” for content stored in The Internet Archive), but rather a direct web URL, then one is looking at something which is online from the live web.

However, this method will not always work on embedded content since the URL may not be shown.

If a separate archive URL appears, this suggests that the content in question has actually been copied to the archive, but this address may also have been generated as part of an attempt to archive the content that was not necessarily successful.

If the question is important, then the first attempt to verify that the content exists in an archived form is to try opening the archive URL for it. If this leads to a separate archived copy of the content, then it is in the archive.

Another option is to study the metadata for the archived web page and look for details on how the web elements, the bits and pieces of the reconstructed web page one is looking at, have been retrieved, and specifically if the content in question comes from an archived copy of any kind. This attempt at verification is only possible if relevant metadata access is provided, as in The Internet Archive (see 8.1.4 Examples from The Internet Archive, p. 141-143).

A final option is to check if the content still exists online by going to the original source as suggested in 7.4 Missing Content Elements.

If the original source does not exist online, or if it has changed visibly from the content found in the archive, then the content in the archive is confirmed to be an archived copy, *provided* that the content does not depend on geolocation.

If the content depends on geolocation, e.g. a local or national weather forecast or other type of local news feed, then a visible difference may be caused, not by content being archived but by the user accessing it directly versus looking at in via archive servers placed in another country. But if the content is the same whether one looks at it from one country or another, e.g. a video, a document, or an image, then the test of online version versus archived version will hold.

7.7 The Archived Web as Data

Another research perspective – than merely revisiting content or looking for earlier versions of content that may no longer be available on the Web – is implied by the amounts of data that are stored and can be retrieved with large-scale archiving. That is, the possibility of distant reading; the study of many resources as data (see 3.4 Relating IT Skills to Project Complexity", p. 40, and 4.1.3 HTML: HyperText Markup Language, p. 62).

Data harvested in crawl by institutional archives are stored in a compressed format which contains all the data retrieved in the crawl; HTML documents, other types of documents, images, videos, audio files, executable files, etc. These compressed files normally have the WARC format, or the newer WACZ format.

WARC stands for "Web ARChive"; WACZ stands for "Web Archive Collection Zipped". WACZ is a newer format developed by the web archiving service Webrecorder [15] which also provides software for replaying content saved in the WARC and WACZ formats[16]. Both file types are compressed formats where all the harvested information for one or more websites is packed, and can be unpacked again, similar to a ZIP file. The unpacked data contains all the information necessary to rebuild the harvested website(s) or web page(s), but the data will not resemble a local harvest with HTML files as described in 9.4 Archiving Websites or Specific Content from Websites unless dedicated software for replay is applied.

Obtaining data from institutional archives can be difficult (it will normally require a special agreement and permission), and large scale data treatment may call for special IT skills, programming and/or equipment at a level that is beyond the scope of this book.

---

[15] https://web.archive.org/web/20240817031358/https://webrecorder.net/2021/01/18/wacz-format-1-0.html

[16] https://webrecorder.net/tools#replaywebpage (This URL may or may not continue to work).

However, many kinds of large scale data treatment can be done with data collections with relatively simple and accessible methods; e.g. examining language such as counts of specific expressions, or extracting specific files types for various types of analyses, such as statistics.

The institutional archives mostly reflect a tradition where web archiving started with the purposes of preserving content and offering the possibility of revisiting it.

In an article on combining different datasets, Niels Brügger explains the relevance of supporting archived web content as data like this:

“To cater to an increasing interest among researchers who are more familiar with the digital humanities than with web archives, it is pivotal that web archives as data be made available as such, and that information about the provenance of the collections, including documentation on curatorial choices and technical decisions regarding collecting, preserving, and extracting the data, become transparent. This will help researchers to get an in-depth understanding of the specificities of the archived web as a unique form of reborn digital material (Brügger 2021, p. 166).”

As the amounts of data have grown, and researchers have expressed interest in the archived resources not just as content but also as data, the institutional archives tend to orient themselves more in such a direction by adding access to metadata or analytical tools in their own interfaces. Examples of such tools are given in 8.1.4 Examples from The Internet Archive, and 8.2.5 Examples from Netarkivet.

8 Existing Web Archives

Takeaways

♦ There are many institutional web archives, most notably The Internet Archive which is open to the public, and national web archives.

♦ Institutional web archives have different strategies for harvesting as well as different policies for access.

♦ The Internet Archive has a powerful service for preserving web pages at the user's request ("Save Page Now").

♦ Access to metadata, and especially provenance data, is crucial for closer inspection of content if one needs to know exactly what is archived, or refer directly to specific content.

♦ Referencing web content via web archives is a more stable solution than giving the original URL and retrieval date. If one is referencing from archives without public access, or from different archives, the PWID standard is the best unified solution.

♦ Subchapter 8.1.1 Author's Note addresses a problem that may occur depending on the browser used for visiting The Internet Archive.

There are many existing web archives where one may find and search for content. One may for example find thematic collections created for a specific topic or purpose, and made public or accessible to researchers upon request by those who created the archive.

But the most interesting for broader purposes are the institutional archives that conduct systematic archiving of large amounts of web content in order to preserve cultural heritage as historical artefacts, and this is the kind of web archives that will be treated in this chapter.

The most notable web archiving initiatives in a broad sense are The Internet Archive which is the world's first, largest and open web archive, harvesting web content since it was established in 1996, and the various national web archiving initiatives which all harvest systematically at large scale, with a primary focus on web content from the respective countries' top domains (.dk, .nl, .fr, .au, etc.).[17]

But there are also more specialised web archives, e.g. the End of Term Web Archive established in 2008, which specialises in harvesting and preserving governmental web content from the Legislative, Executive, or Judicial branches of the government at the end of presidential administrations, or the Library of Congress Web Archives which provides curated collections for a range of topics such as events or international politics.

With many institutional web archives in existence, this chapter will offer a closer look at two; The Internet Archive which is the world's oldest, largest and open web archive, and Netarkivet, the national Danish web archive.

The chapter will follow up with advice on finding other archives with mention of a few significant examples, as well as a few noteworthy resources.

[17] Websites relevant to a specific nation may also be retrieved from .net, .com, or other top domains. See for example the explanation of "danica" in 8.2 Netarkivet, p. 147-148.

8.1 The Internet Archive

The Internet Archive (https://archive.org) is the world's first and largest web archive. It was founded in 1996 by Brewster Kahle as a non-profit organisation.

The aim has been that of a universal library. That is, a library striving to collect and preserve copies of all library content, e.g. copies of all books in existence – or all websites and -pages in the case of The Internet Archive. Categories such as books, movies, documents, software, etc. have been added the The Internet Archive later, but here the focus will be on the archive's original and most well-known purpose of archiving and preserving the Web in its entirety to the extent that this could be done.

What could be done depended on two things: What was technically possible, and how robots.txt should be handled. See 5.1 Challenges for Web Crawlers for explanations.

Since The Internet Archive was not created and based on a legislation, they had to decide how they would handle it when websites had a robots.txt file designed to deter crawlers. As an ethical decision rather than a legal one (robots.txt specifications are not legally binding), they decided to abide with requests for websites not to be harvested. This started changing in 2017, where The Internet Archive started a harvesting policy for better support of completion and accuracy:

"We see the future of web archiving relying less on robots.txt file declarations geared toward search engines, and more on representing the web as it really was, and is, from a user's perspective (Graham 2017a)."

It is unclear how this may affect content that was originally online before 2017, but it is likely that some content from before 2017 will be missing because it was not harvested due to robots.txt rules.

The harvesting as well as the reconstruction or "replay" of archived content is managed by the WayBack Machine, developed for and by

The Internet Archive. This program is used in various setups or modified versions at many institutional web archives, including the national Danish web archive, Netarkivet, with two different versions in 2024, 8.2.3 OpenWayback and 8.2.4 SolrWayback.

The Internet Archive is open to the public and as of 2024 serves millions of people on a daily basis, and contains over 835 billion web pages.[18] Since 2017 it has supported access to provenance data for the harvested content, because this is important to researchers and others who may need precise insights in where the content came from, how it was reconstructed, and what exactly is in the archive (Graham 2017b).

Examples of various forms of access to provenance data are given in 8.1.4 Examples from The Internet Archive, and 8.2.5 Examples from Netarkivet.

Because of the heavy load on the service, server or functionality downtimes are not unusual, but such errors are usually corrected quickly. Looking for a specific web page may yield no results due to a temporary malfunction, but in case of a malfunction an existing archived copy may usually be found by trying again a couple of hours later, or the next day. The “Save Page Now” service may also fail due to overload or temporary malfunctions, and in that case should simply be retried later (see 8.1.2 The ‘Save Page Now’ Service).

The Internet Archive offers sign-up for user accounts, a service that has hitherto been and may remain free. An account is not necessary in order to visit and use the web archive, but it does give access to several functions, e.g. creating lists of favourites. A free user account also adds dramatically to the functionalities of The Internet Archive’s “Save Page Now” service.

Before proceeding with a closer look at archived web pages in The Internet Archive, a basic glitch which was encountered should be mentioned:

[18] https://web.archive.org/web/20240831134647/https://archive.org/about/

8.1.1 Author's Note

Whether the WayBack Machine Toolbar with a timeline and access to provenance data[19] is shown at the top of archived web pages may depend of the browser used for visiting The Internet Archive.

While creating screenshots of archived web pages from The internet Archive in the summer of 2024, a problem was encountered: The WayBack Machine Toolbar with a calendar view, and the "About this capture" function that should be at the top of an archived web page was not shown in the browser, Mozilla Firefox.

By searching for "internet archive unhide wayback toolbar" the first finding was that it is necessary for full WayBack Machine functionalities that JavaScript is enabled in the browser. However, following directions for this verified that it already was enabled in Mozilla Firefox.

Accessing The Internet Archive in the browser Google Chrome, the toolbar was shown, and screenshots were created using that browser.

A Reddit discussion delivered a somewhat unsatisfactory answer: At the time of making the screenshots the WayBack Machine functions were simply unable to be shown in some browsers while readily available in others. The Reddit page is not referenced here since it is a discussion between individuals and may contain person sensitive information.

Some Reddit users speculated that there might be a software conflict, or an error on the side or The Internet Archive, or that a large scale hacker attack in May 2024[20] might have caused these problems.

The problem persisted for several months. It was solved in September 2024. But something similar may occur again, and not necessarily affecting the same browsers.

---

[19] Discussed in 8.1.4 Examples from The Internet Archive, p. 140-142.

[20] More information on the DDoS attack in Freeland, 2024.

Whatever the explanation might be this occurrence demonstrates why having more than one browser is important, and specifically that visitors to The Internet Archive may have to switch to another browser if the toolbar is not visible, or if other expected functionalities fail.

8.1.2 The 'Save Page Now' Service

When searching for a specific web page visitors to The Internet Archive may sometimes encounter, that the page is not in the archive. If the page is online, The WayBack Machine will offer to attempt to add it to the archive.

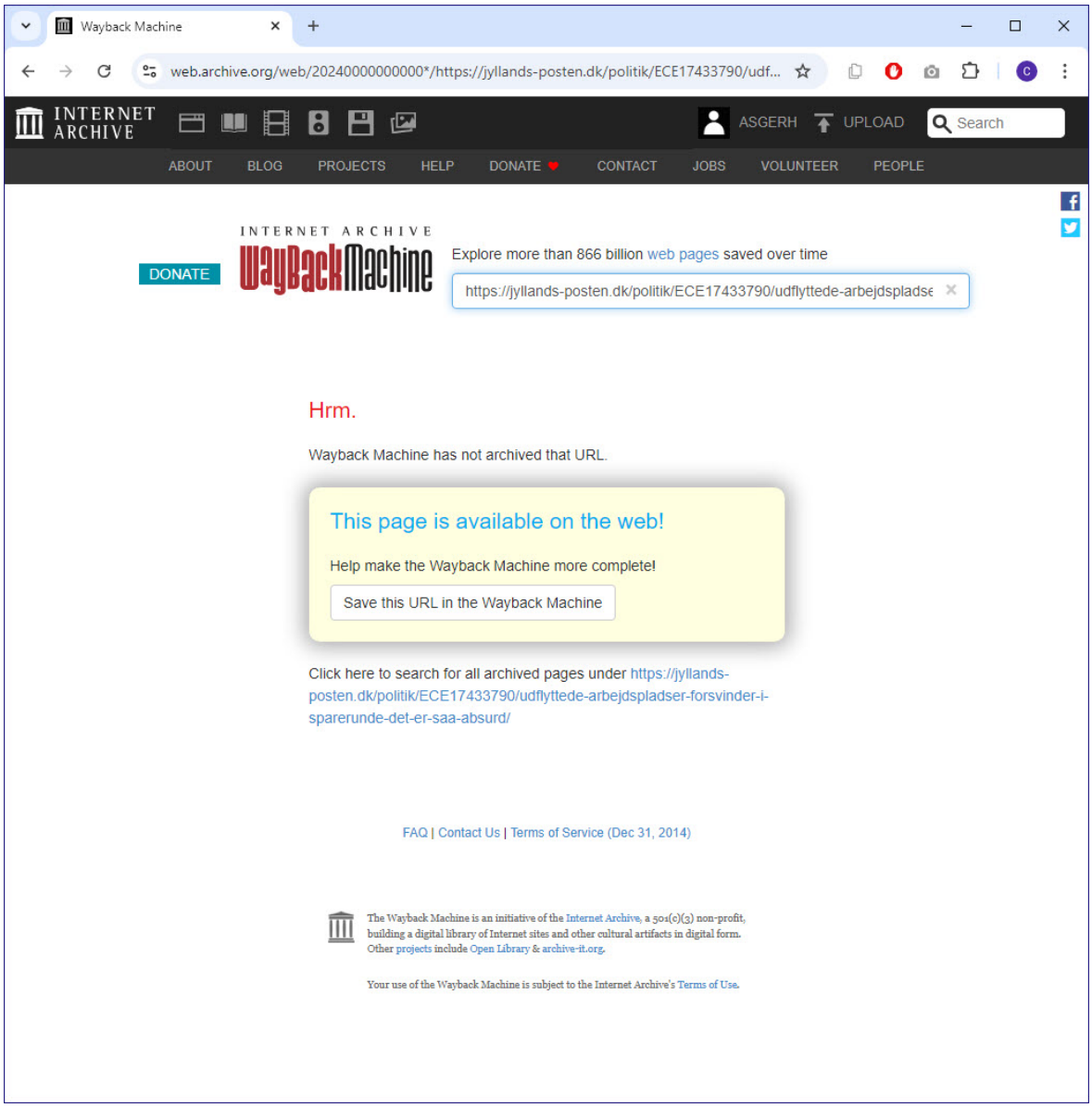


Figure 21: The offer of attempting to save a page that is on the live Web but not in the archive.

Whether the page will be saved by clicking the “Save this URL in the Wayback Machine” depends on whether this is technically possible, as described in 5.1 Challenges for Web Crawlers.

The option of saving a page directly from the live Web is also available directly. The service is aptly named “Save Page Now”. It presently resides in the “Web” section of The Internet Archive, but changes may occur.

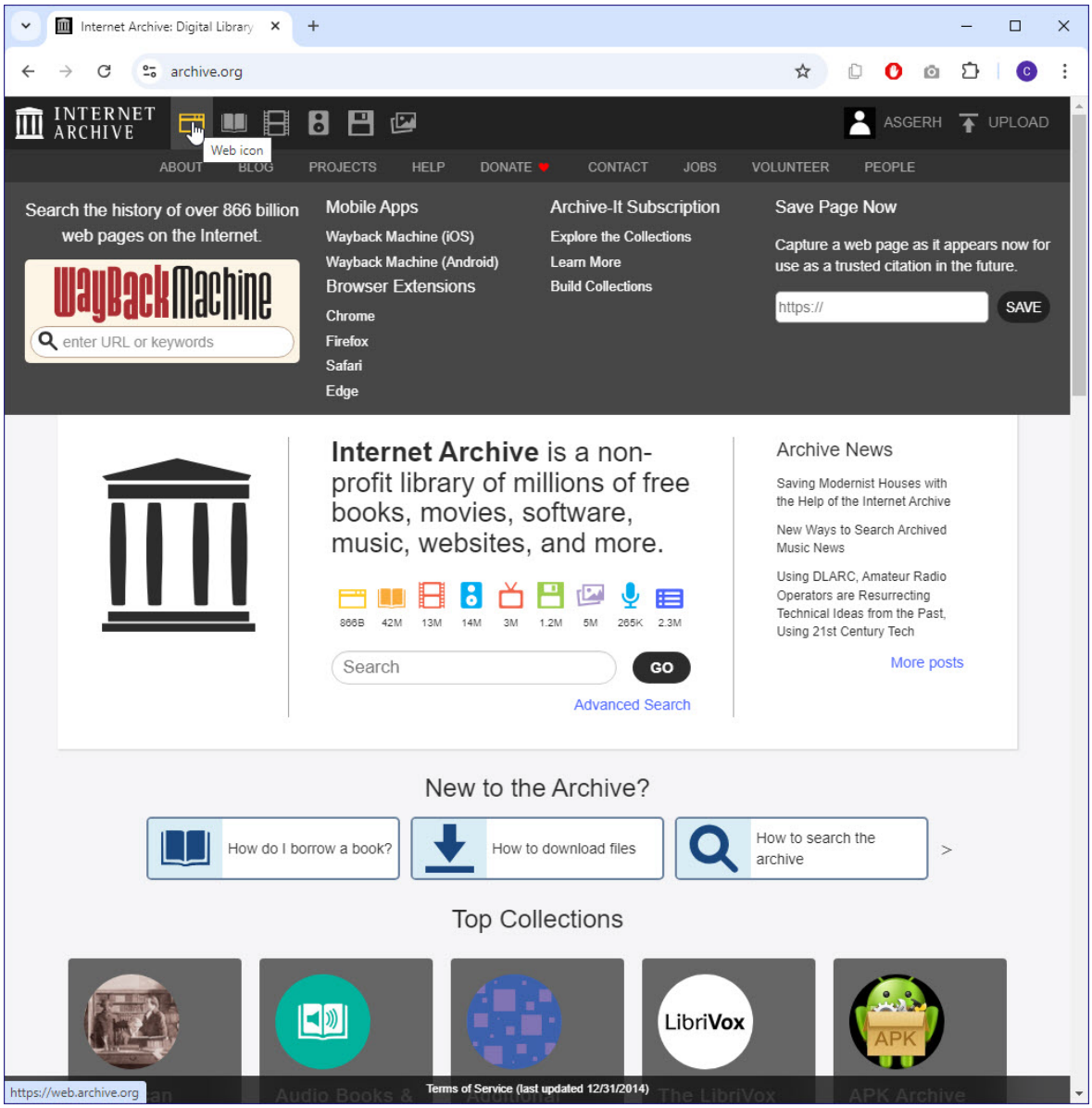


Figure 22: The Internet Archive’s "Save Page Now" service located in the Web section, top right.

If used without a user account, the function will work very similarly to the aforementioned offer of saving a page that is not found in the

archive, but is online. It will attempt to save the page and if the process is successful, the saved page will load in the browser, shown in The WayBack Machine.

But it is worth noting that the function becomes more advanced if used when logged into a user account:

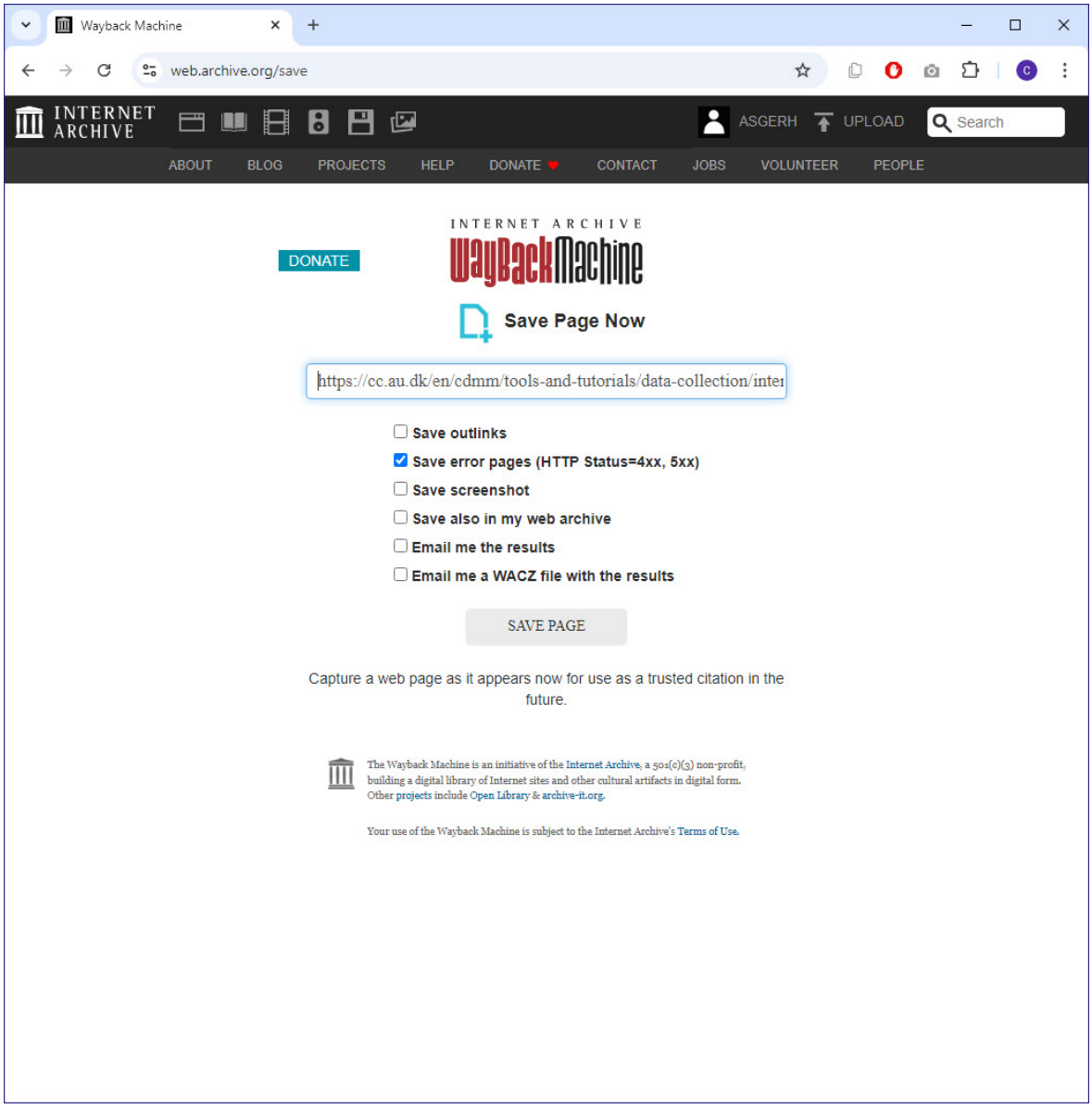


Figure 23: The Save Page now service for registered users.

Users are now offered extra functions which deserve a short explanation (remembering that the functions may change, or more functions may be added). The functions are also described on the list shown on the web page, the details will be shown on mouseover.

- Save outlinks: Will not only attempt to save the page from the specified URL but also include other pages that the specified page has links to. This will not result in a complete crawl of the website where the page resides, but it may result in many additional pages, included those in the website menu if this is also shown on the page. It will often be possible to save an entire section of a website with a single request by using this option.

- Save error pages. Will save 404 messages and other types of messages saying that a page cannot be accessed, e.g. if the website server is down, or the page no longer exists.

- Save screenshot: Will save a screenshot of the page when accessed. The screenshot option comes with the disadvantage that a screenshot will be taken from a direct visit to the page, including any kinds of pop-ups such as notifications requesting cookie permissions. Pop-ups may disturb a screenshot, making it less useful than one taken manually after closing them. See example below.

- Save also in My Archive: The copied page will be added to a personal collection in the user's account, where it will be available directly (it will also be available to anyone searching for it, "My Archive" merely serves as personal bookmarks).

- Email me the results: Will send a notification with direct archive links and a report on what was saved when the saving process is done.

- Email me a WACZ file with the results: Will send a download link to the user for the newly harvested data compressed in a WACZ file (see explanation on the WARC and WACZ file formats in 7.7 The Archived Web as Data).

When a save has been started the user can follow the process until done, or wait for a notification on the web page after the request has been completed – or as an email if this was requested from a user account.

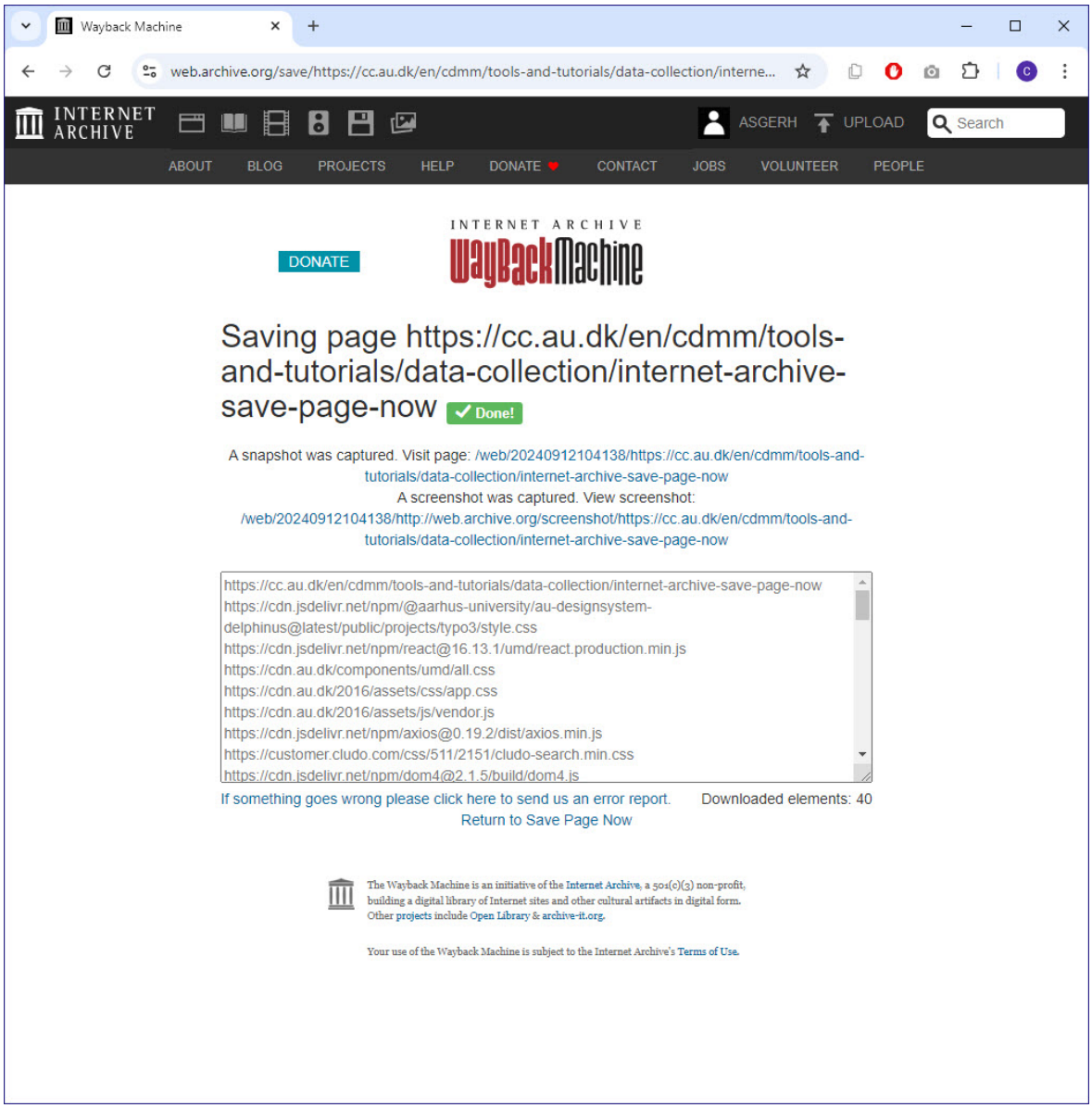


Figure 24: The report after saving a web page.

The report contains a direct Internet Archive URL for the newly saved copy.

IMPORTANT: When inspecting the newly saved copy by opening it, it should be verified that the timestamp in the archive URL (see 8.1.3 Internet Archive URLs) is the same as the one given for the archived

copy in the report. If not, then the newly saved copy has not yet been added to the large index of the WayBack Machine and the URL for the newly saved copy should be reinspected later, after a couple of hours or the day after. The Internet Archive's WayBack Machine will always lead the user to the closest existing copy to the one requested, as described in 7.6.1 Time Jumps in the Content, p. 107-108. If a newly archived page has not yet been fully registered, the WayBack Machine will not be able to locate it, and the closest available copy will be shown instead.

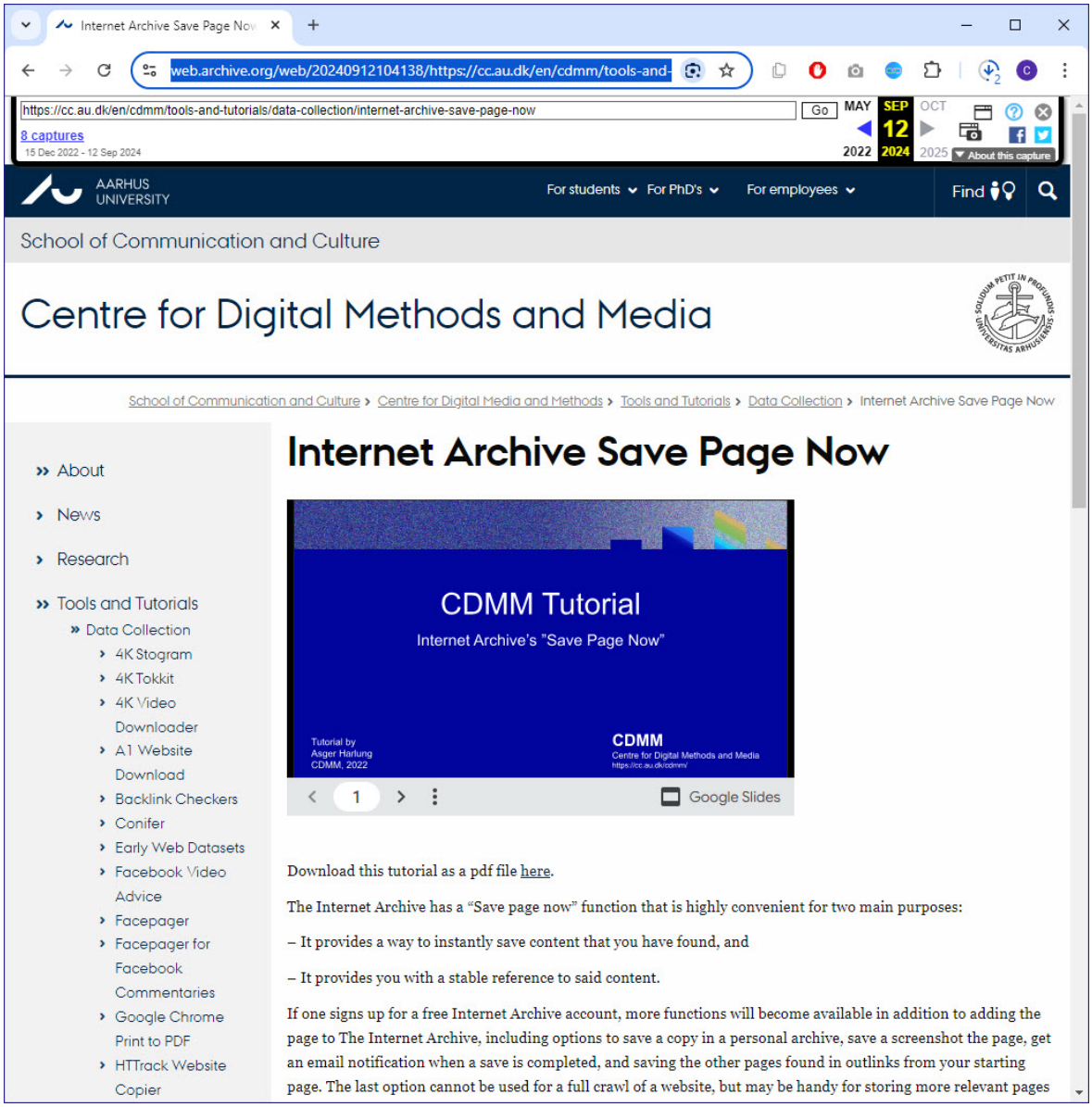


Figure 25: The saved capture, on the URL given in the report on Figure 24.

In the inspection of the copy above, it is verified that the address line given from the “Save Page Now” leads to a satisfying copy of the newly archived page. The timestamp in the archive URL (highlighted in the screenshot) is the same as the one given in the “Save Page Now” report:

https://web.archive.org/web/20240912104138/https://cc.au.dk/en/cdmm/tools-and-tutorials/data-collection/internet-archive-save-page-now

This means that one now has a copy of the online page from the exact time that the copy was made. The archived copy can serve as a stable reference to the page at this point in time, rather than the traditional way of referring to the original URL with a retrieval date (see also 8.3 Referencing from Web Archives).

It should be noted that in rare cases The Internet Archive may hide a website upon the owner’s request.[21] The exact policy for this is not publically available, but it means that even an archived copy may disappear from The internet Archive. But although this can happen on rare occasions, an archived copy at The Internet Archive is still a much more stable copy than any online reference.

Please refer to the next subchapter, 8.1.3 Internet Archive URLs, for a breakdown of the archive URL structure.

The screenshot option, if chosen, may not be as useful as one might wish. The screenshot will capture exactly what appears when first entering a page, and this will often include cookie consent pop-ups or overlays that can disturb or entirely hide the page.

This is the screenshot created in the example from the previous figures:

[21] https://web.archive.org/web/20240922105037/https://help.archive.org/help/using-the-wayback-machine/

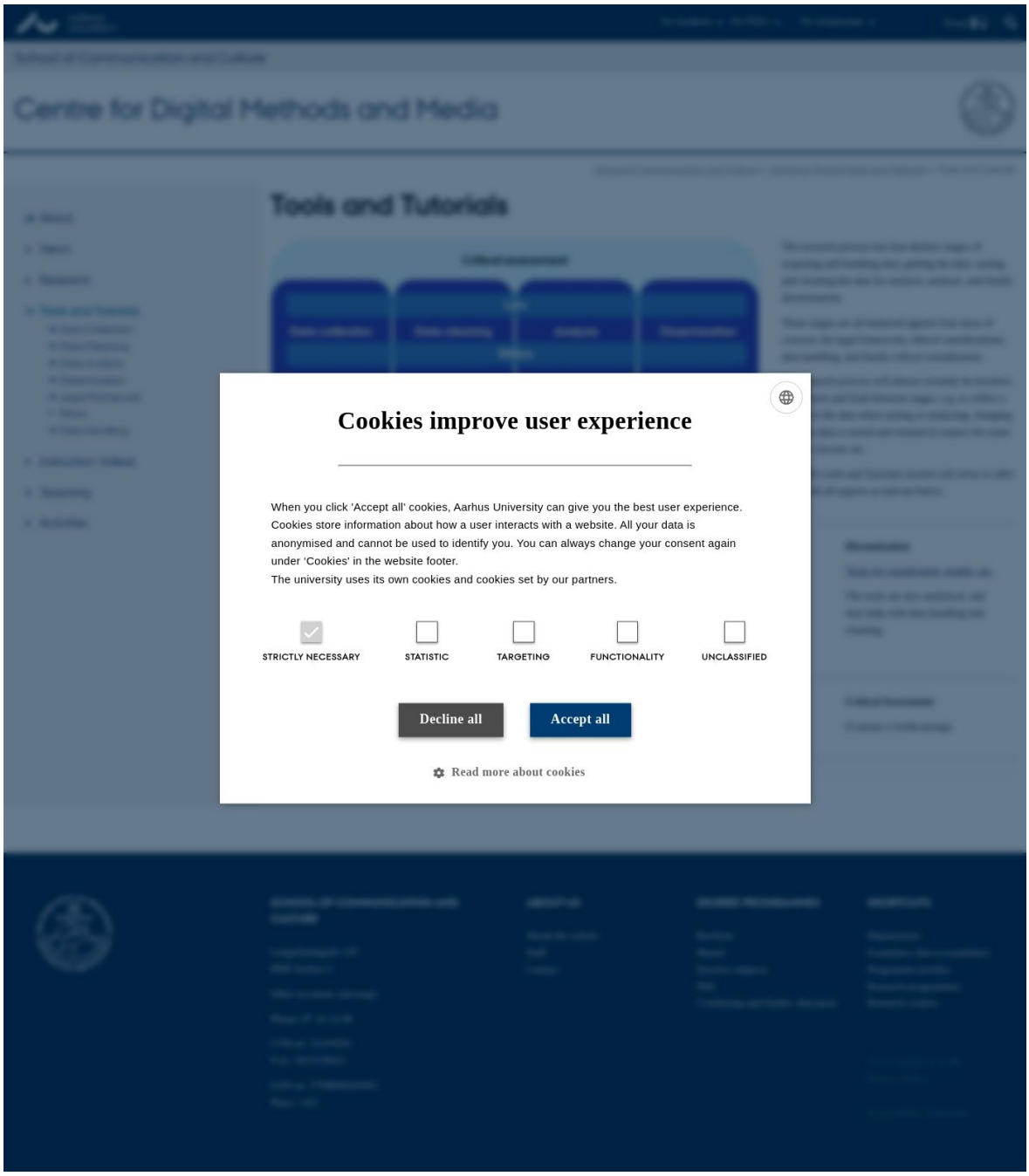


Figure 26: Cookie consent pop-up included in a requested screenshot.

The Internet Archive offers WayBack Machine browser add-ons for several major browsers [22]. The add-ons provide the possibility of requesting an archive save of a page with one click from the browser, with the reservation that the “Save Page Now” will not work for pages

[22] Alexis Rossi (2017): If You See Something, Save Something; https://web.archive.org/web/20240731135122/https://blog.archive.org/2017/01/25/see-something-save-something/

that cannot be saved for technical reasons. The add-on also allows the user to check for the oldest or newest archived version of a live web page, the go directly to The Internet Archive and see how many copies from the same URL exists, etc.

The add-on is identified as "Wayback Machine by Internet Archive", and may be found by searching for "Wayback Machine" in the extensions catalogues for the browsers where it is available. Presently (2024) the add-on is available for Google Chrome, Mozilla Firefox, Safari, Edge, and also as an app for Android and iOS via their respective app stores.[23]

As mentioned in 10.6 Browser Extensions, any browser add-on may occasionally loose functionality after a browser update. Should this happen with the The Internet Archive's Wayback machine add-on, there are two possibilities; either to wait for an updated version (the user must check manually for this by going to the extensions catalogue, or looking in the Web section of The Internet Archive, https://web.archive.org/), or change to another supported browser where, hopefully, the add-on may still work.

The primary advantage of using a browser add-on is the option to capture pages with just one click, with the disadvantage that a visible report is not generated or emailed. The button "Display a Calendar View of Archives" will open a new tab at archive.org showing the calendar view for the relevant page.

The add-on needs to be logged in to an Internet Archive user account in its settings order to get full functionalities (see the introduction in 8.1 The Internet Archive).

[23] https://web.archive.org/web/20240807000315/https://web.archive.org/

8.1.3 Internet Archive URLs

URLs for web page copies preserved in The Internet Archive consist of three items:

- An address for The Internet Archive where the copy is located, e.g. https://web.archive.org/web/,
- A Coordinated Universal Time (UTC) timestamp consisting of year (four digits), month (two digits), day (two digits), hour (two digits), minutes (two digits), and seconds (two digits),
- The original address for the web page as located on the Web.

In the previous subchapter, the "Save Page Now" function was used to create a copy of the web page, https://cc.au.dk/en/cdmm/tools-and-tutorials/data-collection/internet-archive-save-page-now.

The resulting copy had the archive URL:
https://web.archive.org/web/20240912104138/https://cc.au.dk/en/cdmm/tools-and-tutorials/data-collection/internet-archive-save-page-now

…which consists of the following three items:

- https://web.archive.org/web/ – signifying that the copy is located in The Internet Archive's web collection,
- 20240912104138/ – signifying that the copy was taken at 10:41:38 hours on September 12, 2024, and
- https://cc.au.dk/en/cdmm/tools-and-tutorials/data-collection/internet-archive-save-page-now – which was the original address for the web page.

The archive URL contains all the information needed for referencing a web page as it was when it was archived. Please refer to chapter 8.3 Referencing from Web Archives for a discussion on how to refer to content found or saved in web archives.

8.1.4 Examples from The Internet Archive

Please notice that the purpose of the examples given here is to convey an impression of what one may find in a web archive. Any functions named or shown may change, and new functions will more than likely be added.

When entering the main page for The Internet Archive one is immediately greeted with a search line for the WayBack Machine. This is the The Internet Archive's core function for retrieving and representing archived web pages.

One can start a search by using search terms, or enter the URL or the domain for the content of interest. As with modern browsers, domain names and page addresses can be entered without the full URL (see 9.3 Copying URLs).

It should be noted that (so far, until and including 2024) search terms or keywords can only be helpful for finding the front pages of websites, and do not work as a full text search. The Internet Archive hopes to be able to offer full text search at some point in the future.[24]

In the following examples, a search is done for netlab.dk, a Danish domain that was active from 2012-2022.

[24] https://web.archive.org/web/20241204132356/https://help.archive.org/help/using-the-wayback-machine/

Figure 27: Starting a search in the WayBack Machine.

The results are given in a calendar view starting at the present year which is highlighted in yellow as shown in figure 28.

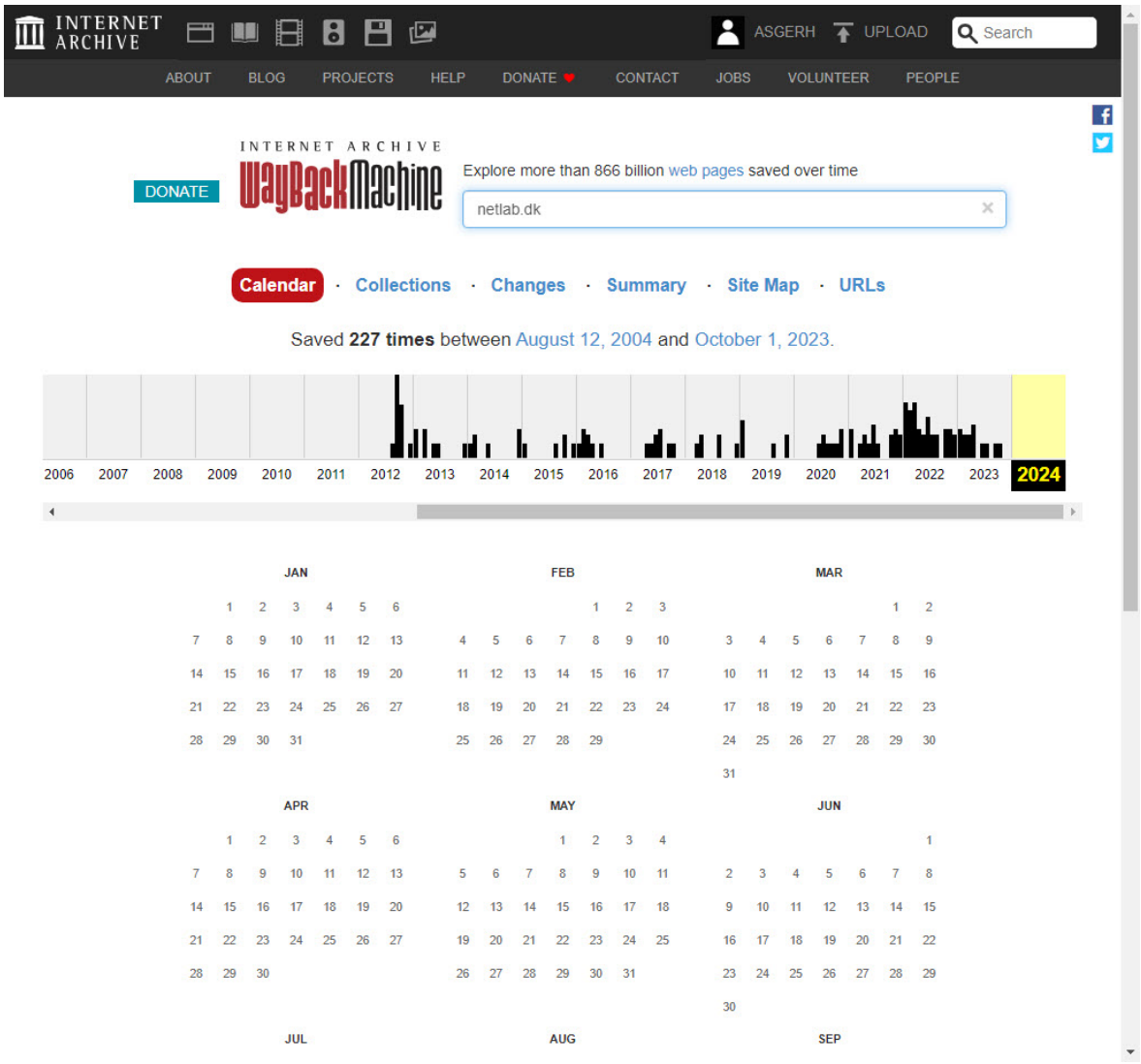


Figure 28: Calendar view of search results.

The bars shown for each year indicate amounts of archiving by month. In this example a closer look will now be chosen for 2022. Clicking that year in the timeline results in the months shown below changing with blue and green circles on some dates.

Some circles are larger, indicating that there were more harvests of the same page on those dates:

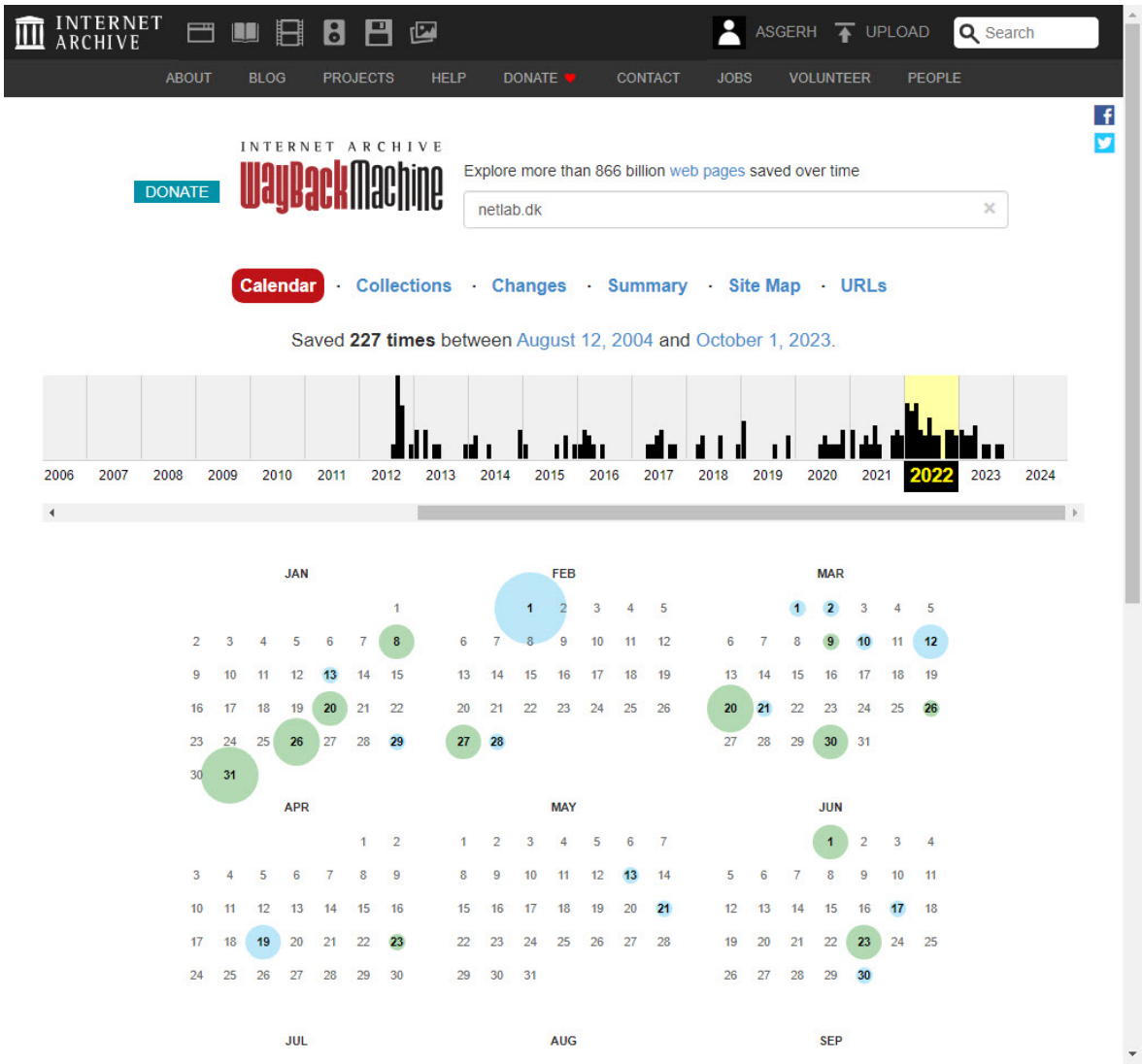


Figure 29: Calendar view of archived copies of netlab.dk in 2022.

Sometimes the calendar view will only have blue circles. They indicate that the page in question was targeted directly for harvesting, and that harvesting took place with no error notifications which usually implies that there will be a good copy of the page.

When circles in other colours are shown, an explanation can be found at the bottom of the calendar overview page:

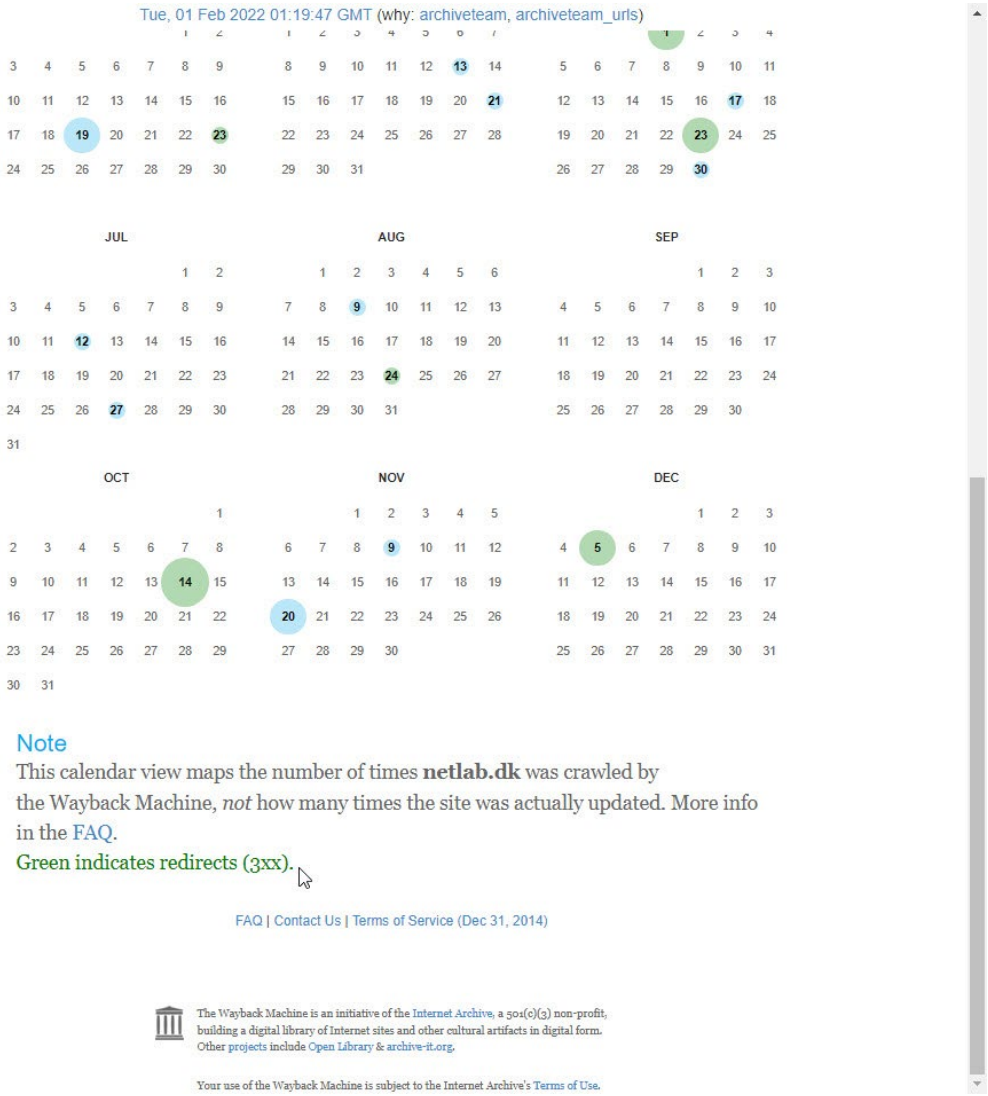


Figure 30: Bottom of calendar view for 2022, with explanation of green circles.

The explanation given is that green circles indicate redirects with an HTTP code in the range from 300 to 399. The specific cause for a redirect may differ, but the implication is that the page was not harvested intentionally, and is thus a by-harvest from a process where it was not targeted deliberately. Other possible colour codes are orange, indicating that a page was not found, or red, indicating that a server error was encountered. In such cases the archived pages will be copies of the error messages that were encountered.

If one needs a good copy, going to a direct harvest (blue) may be the safer choice, but if one needs something from a specific time where the closer versions are from by-harvests, then these copies may also be of good quality.

Above the timeline in the calendar view there are five different alternative menus for exploring the archived content for the web page in the search (here, netlab.dk).

All these offer different ways to get an impression of the collection, by showing metadata on where the content came from, which files types have been harvested, subpages in the harvest, etc. These data overviews can be interesting and may offer some background or inspiration.

- Collections – Will list the various ways that the copies may have been added to the archive, e.g. in “common crawls” initiated and planned by The Internet Archive, by Save Page Now requests (see chapter 8.1.2 The ‘Save Page Now’ Service), or in special harvesting projects.

- Changes – Will provide a calendar overview with changes indicated, and a possibility of selecting and comparing two different versions of the web page. This can be useful for doing side by side comparisons of significant changes, but there is a risk of finding only minor changes, e.g. where a picture was re-uploaded to the web page without any change, or where a typo was corrected.

- Summary – Presents a series of graphical overviews of amounts of various file types included in the harvests of the web page.

- Site Map – Shows a graphical overview of archived pages from a website by year, giving an impression of what has been preserved and which pages belonging to the domain one may find in the specified year. This function provides an overview of archived copies for the entire website, also if the initial search was made for a specific page. Pages shown in the graph may be opened by pointing at them and clicking or right-clicking (ctrl+click on Mac OS).

- URLs – Provides a full list of all URLs harvested for the web page in the search, including the separate URLs for all files. The list can be very long and may be divided over several pages.

This is an example of the Site Map view:

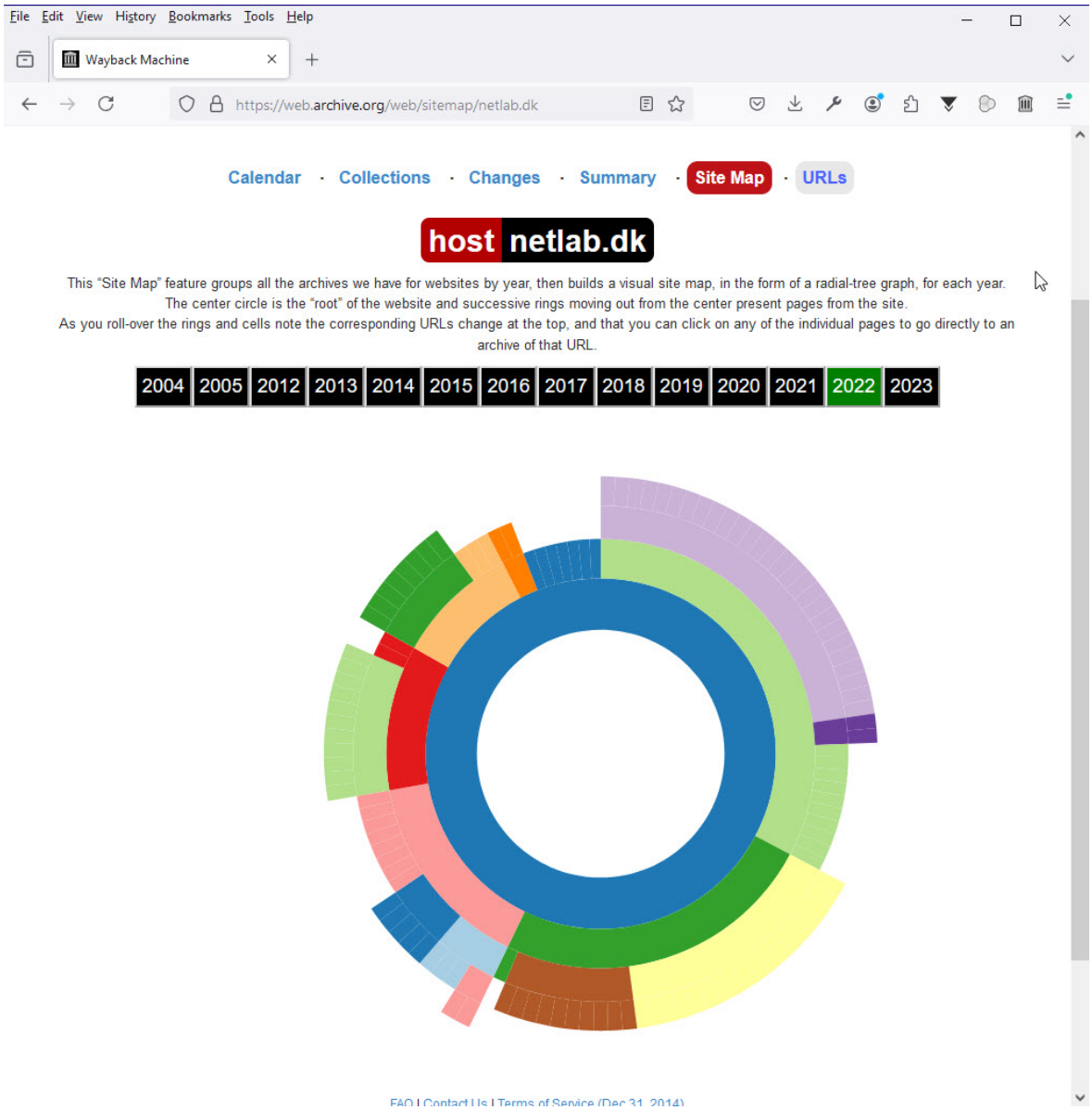


Figure 31: Site Map view of pages harvested for a domain in the specified year, here, 2022.

All these functions are a result of The Internet Archive working towards making metadata more accessible for researchers or others that may take an interest in exploring the nature and provenance of archived data in more depth.

Returning to the calendar view for the domain netlab.dk in 2022, a closer look is now taken at February 1, 2022, where a large blue circle indicates that there will be several copies.

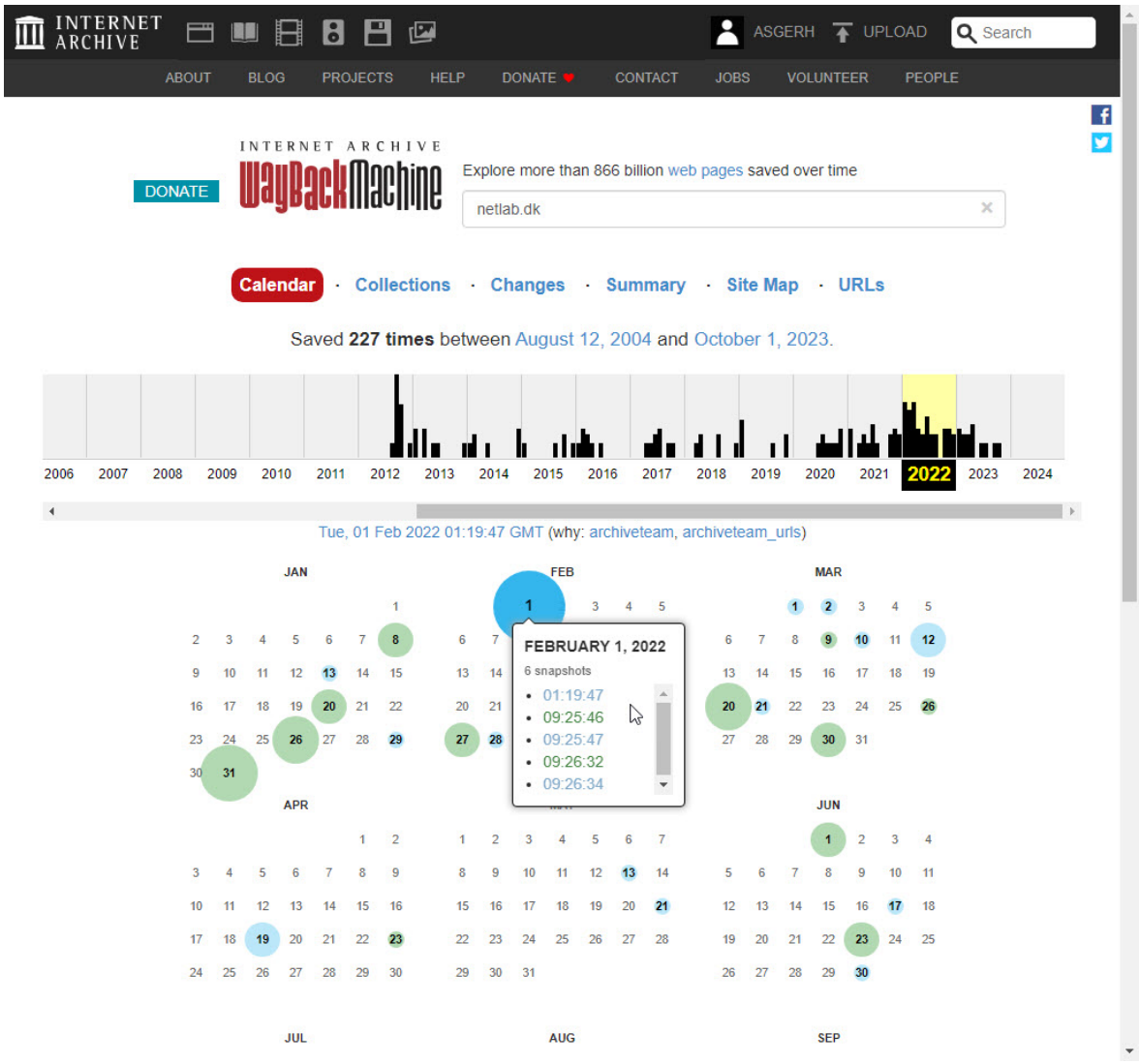


Figure 32: Closer look at February 1, 2022.

Mouseover on a date reveals the copies saved on that date. Some are primary copies, some are redirects. The cursor can be moved to the copy that one wants to look at, and it can be opened by clicking or right-clicking it (ctrl+click on Mac OS).

Opening with right-click and selecting a new tab has the advantage of keeping the calendar ready if one needs to explore or change to versions from other harvests.

Here the first copy from February 1, 2022 at 01:09:47 UTC has been selected and opened in a new tab.

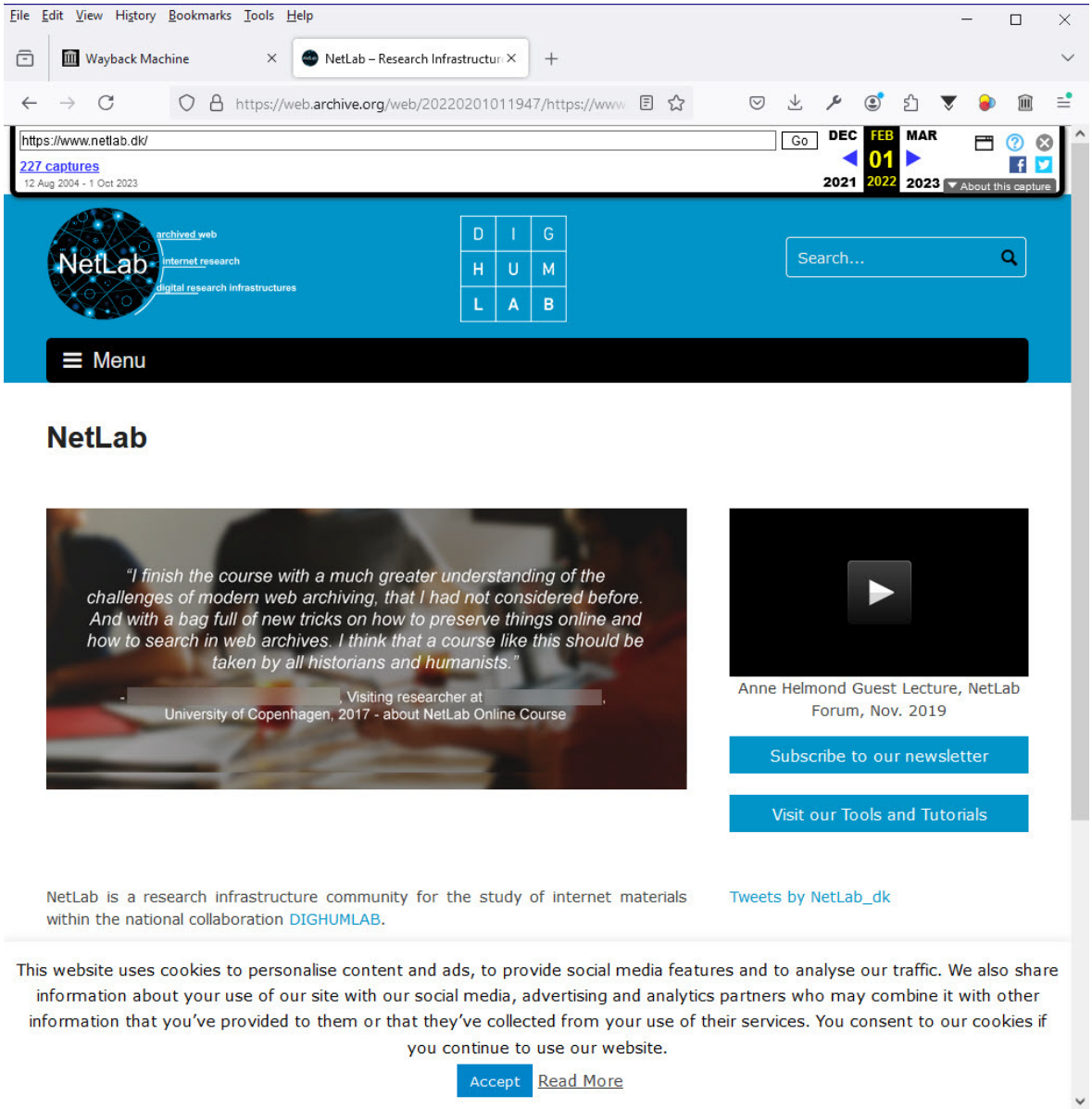


Figure 33: Copy opened in new tab. Cookie consent notification at the bottom. Details blurred by the author.

In the newly opened copy in the example, a cookie consent overlay is shown at the bottom of the page. This was a part of the coding for the website, as it is for most websites and has been since the EU's "Directive on privacy and electronic communications" was applied in 2002.[25]

[25] The law was revised in 2009 and is still in effect; https://web.archive.org/web/20240913180902/https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A02002L0058-20091219

As with the online version of a web page, the cookie consent overlay can be closed by clicking “Accept”. No new cookies will be stored for the archived version by doing so.

The copied version of the page loyally shows the text and images, including a slideshow with changing images that were stored on the page’s own domain and successfully retrieved in the harvest. This can be confirmed by looking at the metadata in the “About this capture” function which is used for a similar purpose in the following example.

However, the archived page also contains an example of something where online leaking may be suspected (see 7.5 The Risk of Online Leaking), namely a video of a guest lecture from 2019 to the right of the slideshow.

If one clicks the video it will load and play, in contrast to the parallel example in 8.2.5 Examples from Netarkivet, p. 161-165 where the archive is isolated from the rest of the Internet. But as described in the chapter on online leaking, the video can otherwise be drawn in and shown on the archived web page from an external online source on the Web, via an embed code.

As also mentioned in the chapter on online leaking, there are two primary ways to determine if online leaking is taking place when looking in an institutional web archive (unless it is already known to be protected against online leaking, as is the case with Netarkivet). One is to inspect the HTML code for the archived web page, but the easier and more direct way is to inspect the provenance data for the archived page, if accessible.

Luckily that is the case with pages viewed in The Internet Archive’s WayBack Machine. In the toolbar at the top, one can click “About this capture”, and provenance data will be listed. While they are open, one can search for keywords in the browser, and the provenance list will be included in the search. Provenance data are closed again by clicking “About this capture” once more.

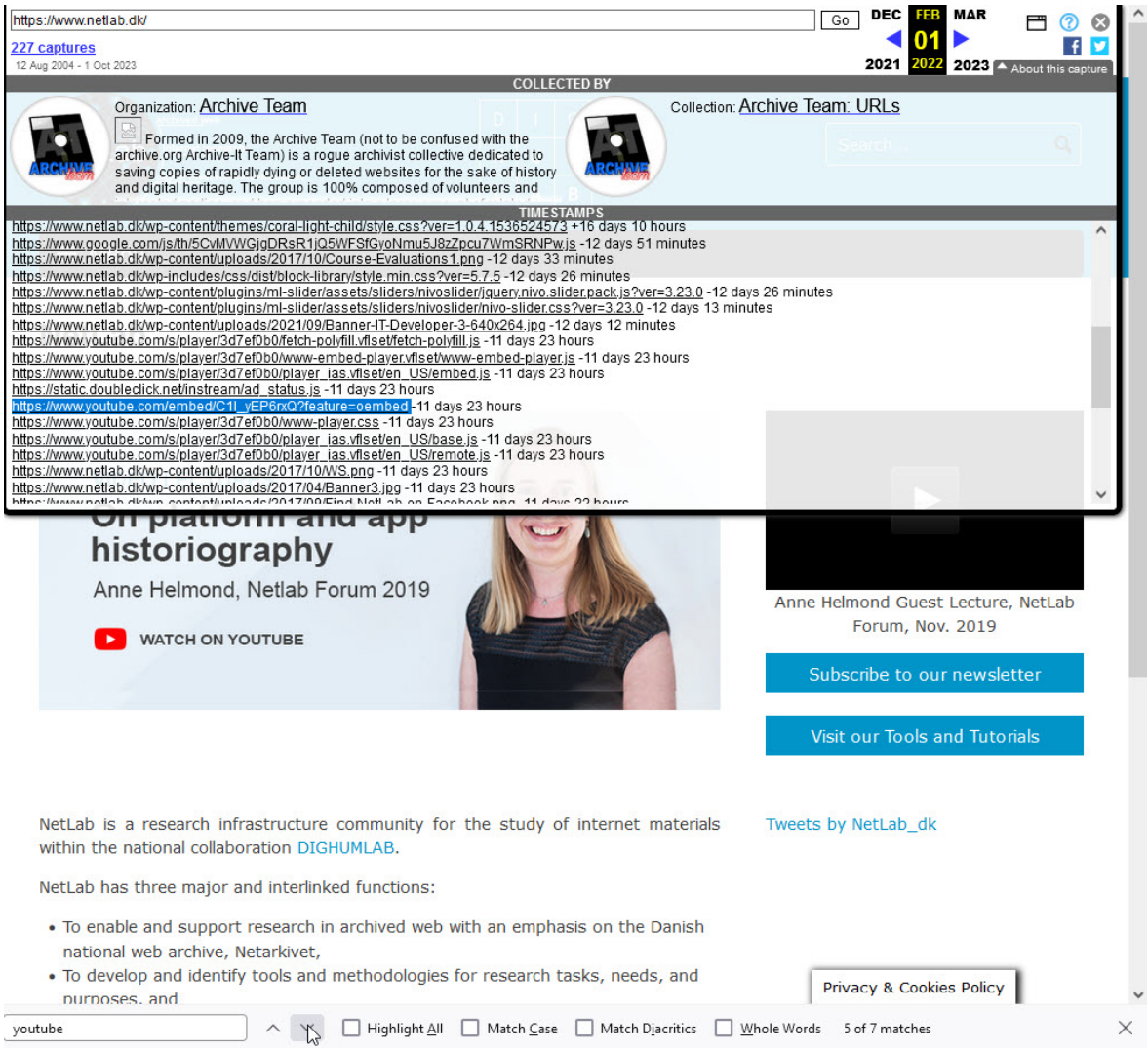


Figure 34: Inspecting the provenance data via "About this capture". Image unchanged by Anne Helmond's permission.

One thing that will likely be striking to most when opening the provenance data is to see the many separate pieces that can go into the reconstruction of a page, and not least the information on how far removed from the capture time for the HTML page itself. In order to save time and storage space, the institutional archives will not harvest new copies of the same files every time a page is harvested. A reconstruction of a page will therefore normally consist of the HTML file at the time of harvesting, supplied with files that were retrieved in other harvests.

It is not uncommon to find that some elements were in fact harvested later than the archived page one is looking at. This is explained in 7.6.1 Time Jumps in the Content, p. 105-106.

The provenance data are helpful for determining exactly which elements are in an archived version of a web page. With the data opened, a search can be conducted for “embed”, or in this case more directly for “youtube”. Several YouTube elements are found, but none of them are video files. The embed itself is found and highlighted in the example (figure 34).

What this means is that the video has not been archived. It can only be played because it is still available on YouTube, and will no longer be accessible if the video disappears from that service. Opening the embed URL in a new tab or window will lead to an archive page with an archive video player calling the video, but not to a video file.

The archived page can be used as an offset for looking around on the entire archived website, but as described in 7.6.1 Time Jumps in the Content, p. 107-108, the user should be wary that this process will usually lead to archived pages that are not from the exact same time as the archived page used as the starting point.

In the final example, the menu on the initial web page is opened, and a page of interest is found. On the next two figures, a mouseover on the page considered for a visit shows an archive URL with the same timestamp as the one for the page where the visit has started:

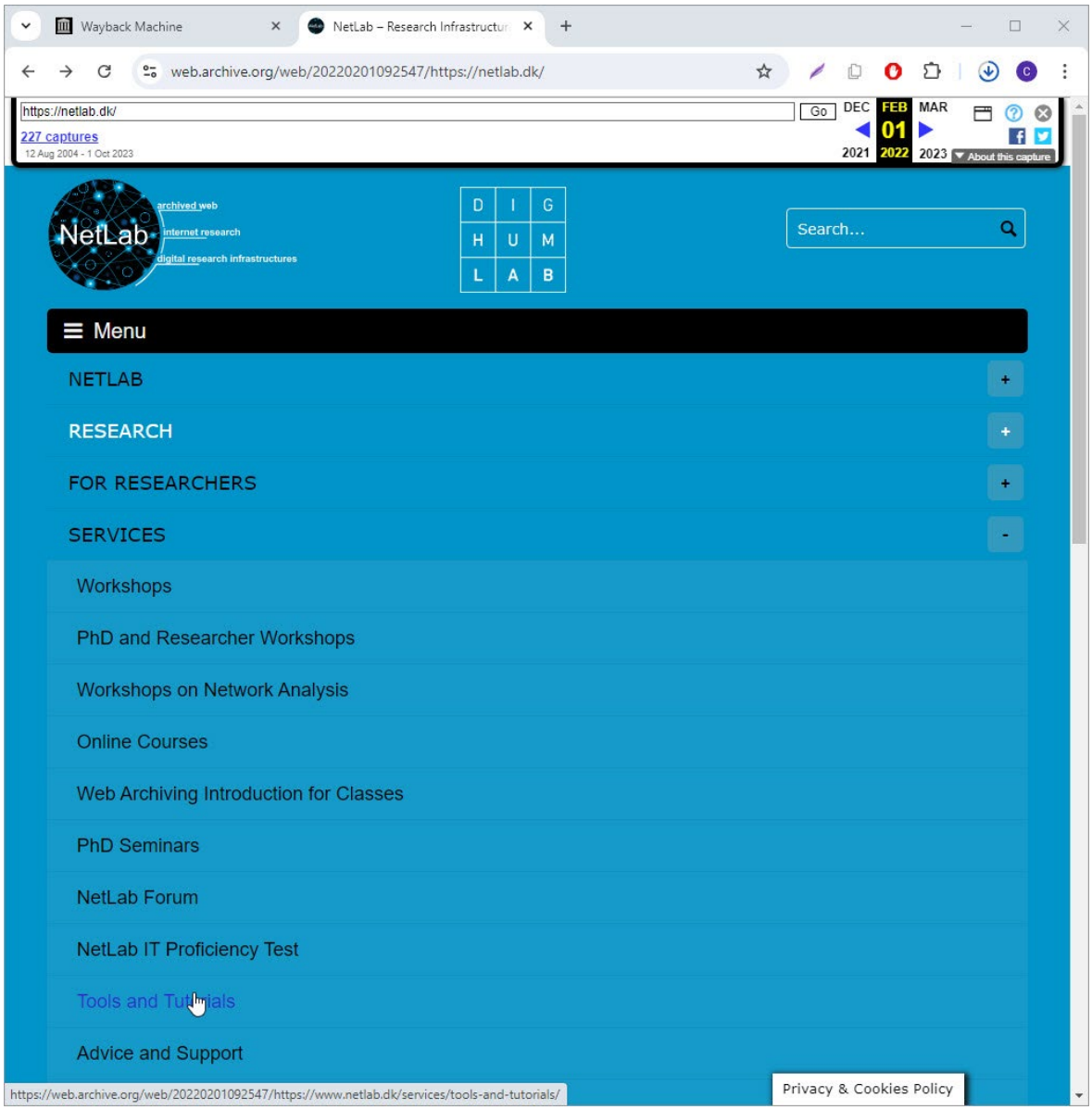


Figure 35: A URL on the archived page has the same timestamp as that for the page presently shown.

Figure 36: A close-up of the previewed archive URL.

However, if opening the link in order to go to the new page, the timestamp changes:

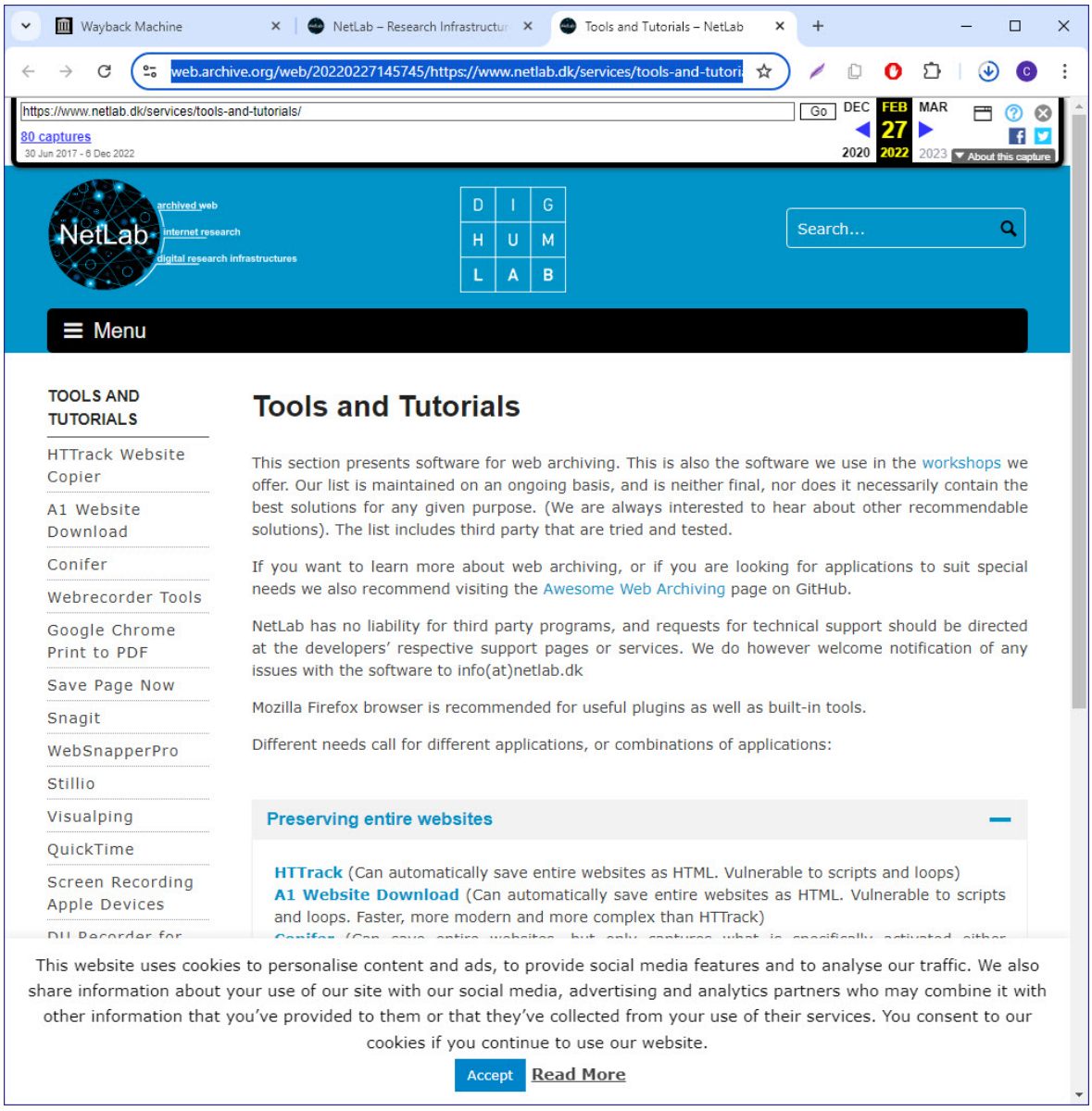


Figure 37: The new page has a new timestamp when opened.

The new page opens with the timestamp, 20220227145745, signifying that one is now looking at a page that was archived on February 27, 2022, at 14:57:45 UTC – whereas the page used as a starting point was from February 1, 2022 at 01:09:47 UTC.

In other words, the archived copy of the new page is removed in time from the starting point by 26 days. This type of time jump means that one cannot refer to an entire archived website from a specific date, but must instead refer to the relevant pages observing their timestamps.

The time jump is caused by the URL from the initial archived web page not being a direct reference, but rather a specification for the WayBack Machine to look for an archived version of the new page as close to the timestamp from the starting page as possible. The process and its implications are discussed in 7.6.1 Time Jumps in the Content, p. 107-108.

The cookie overlay has appeared once again, and may once more be closed by clicking “Accept”.

8.2 Netarkivet

Netarkivet (https://www.kb.dk/en/find-materials/collections/netarkivet) is the National Danish web archive. It was established as a division of the Royal Danish Library in 2005, and has been collecting web content since then.

The Danish Legal Deposit Act (Danish: "Pligtafleveringsloven")[26] is a law that has been in effect since its first version in 1697. It was originally designed for printed materials, and stated that copies of such materials must be provided to the Royal Library. In 2004 content published on the Web was added to the law, detailing that The Royal Danish Library has the right to create the copies.

Another law that impacts the national Danish web archive is The Data Protection Act (Danish: "Databeskyttelsesloven")[27] which effectuates and supplements the EU General Data Protection Regulation ("GDPR", see also 11 Legal and Ethical Concerns, p. 245).

Netarkivet is not open to the public due to the strict legal regulations on personal and person sensitive data in this legal framework, and to the fact that systematic web archiving will unavoidably result in preservation of such data. Access is therefore only granted on application, to researchers or PhD students whose projects make it necessary and relevant.

The content in Netarkivet is primarily harvested from the Danish top domain, .dk, but additionally the archive also targets websites that are categorised as "Danica"; that is, websites in Danish, or created by or for Danes, or in other ways are connected to the Danish web sphere.

---

26 https://web.archive.org/web/20240827114227/https://www.retsinformation.dk/eli/lta/2004/1439

27 https://web.archive.org/web/20240918151416/https://www.retsinformation.dk/eli/lta/2024/289

Since the harvesting process will include links found on web pages, there is also data in the national Danish archive that does not "naturally" belong to the Danish web sphere. Links to international resources found on web pages are also archived to the extent provided by the settings for the web harvests. See also 5.1 Challenges for Web Crawlers, p. 82-83.

The latest published information (as of 2024) on the amount of data stored in the archive is from 2022, where the archive contained approximately 850 TB of data, with over 34 billion objects.[28]

Netarkivet uses four harvesting strategies:

- Broad crawls of all Danish domains, conducted four times annually.
- Selective crawls of web pages that change often or are regarded as being of high importance, such as news media sites, political parties' sites, ministerial sites, etc. The selective crawls are conducted from 12 times daily to once per week.
- Event crawls, based on societal events such as elections, or the Covid-19 pandemic. Event crawls are conducted 2-3 times annually.
- Special crawls on selected topics and/or criteria, e.g. based on requests from researchers.

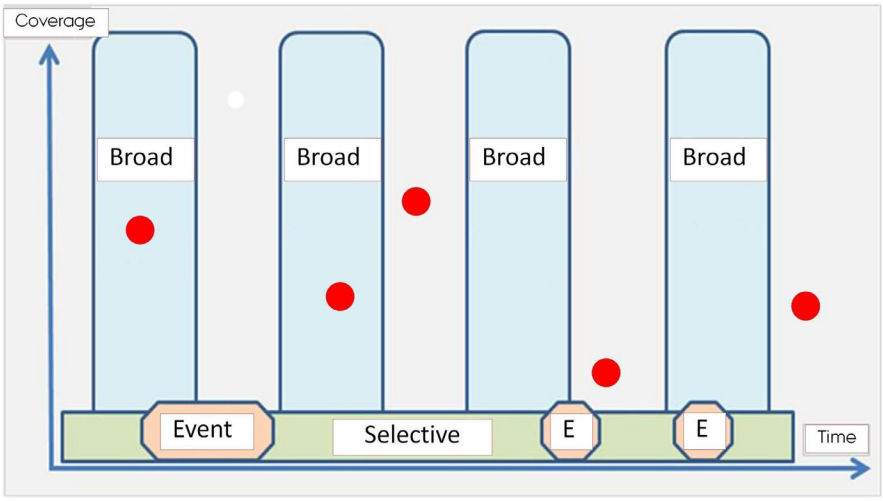
The four types of crawls in combination are designed to secure a broad and consistent coverage of the Danish web sphere.

[28] https://web.archive.org/web/20240829172942/https://www.kb.dk/find-materiale/samlinger/netarkivet

2005 →

Strategies:

- Broad/bulk
- Selective
- Event
- Special



From http://netarkivet.dk/om-netarkivet

Figure 38: Netarkivet's crawl strategies. The red dots represent special crawls.

8.2.1 Access and Appetisers

Netarkivet's home page is the entry point for finding the archive:

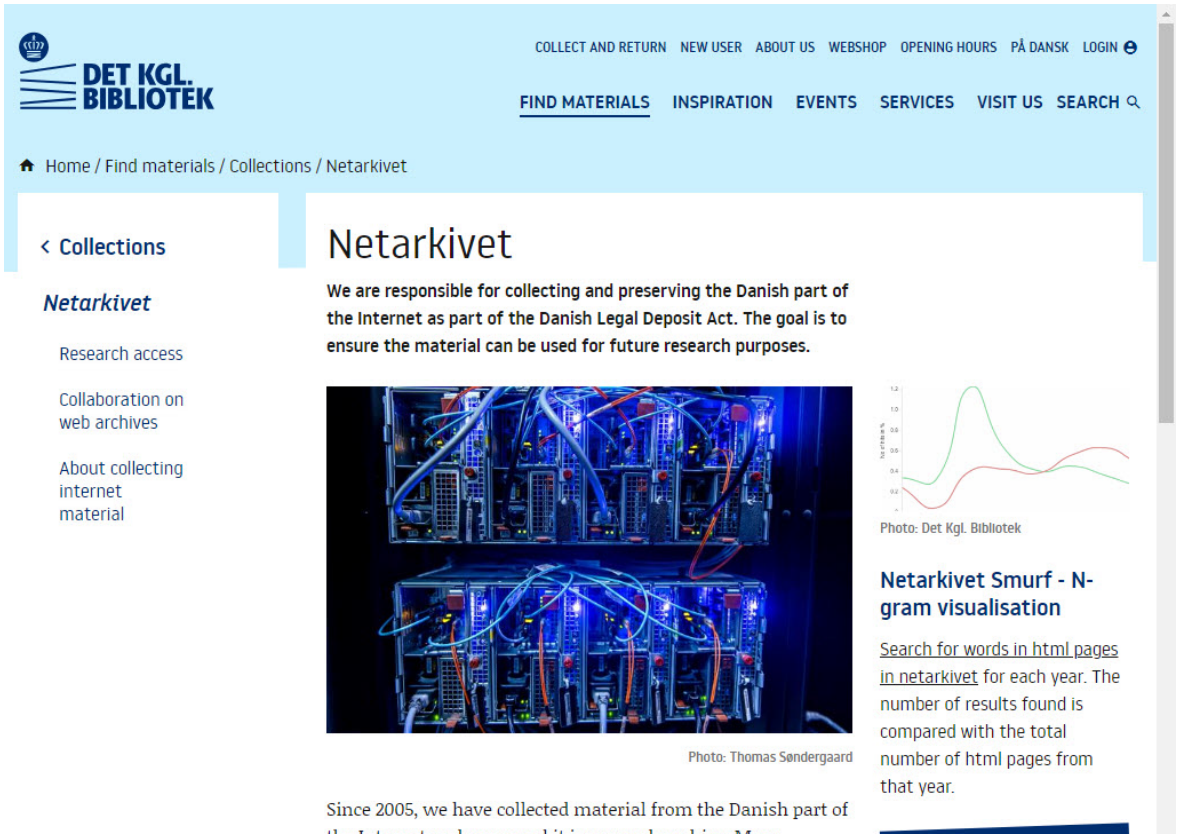


Figure 39: Netarkivet's home page.

It provides introductory information, as well as the conditions and application forms for research access.

It also provides two functions that may serve as appetisers or inspiration for researchers. The functions draw upon data from Netarkivet and returns informational graphs, without providing any direct access to specific data in the archive.

Of the two functions the first, Netarkivet Smurf - N-gram visualisation will be used as an example here. The other, a link graph visualisation showing how a domain is connected to other domains is in development at the time of writing this, in 2024.

In the N-Gram visualisation users can enter a maximum of two keywords that will be searched for in the entire archive. The function will return graphs, showing statistical curves of how many percent of archived web pages were found to mention the search terms over time.

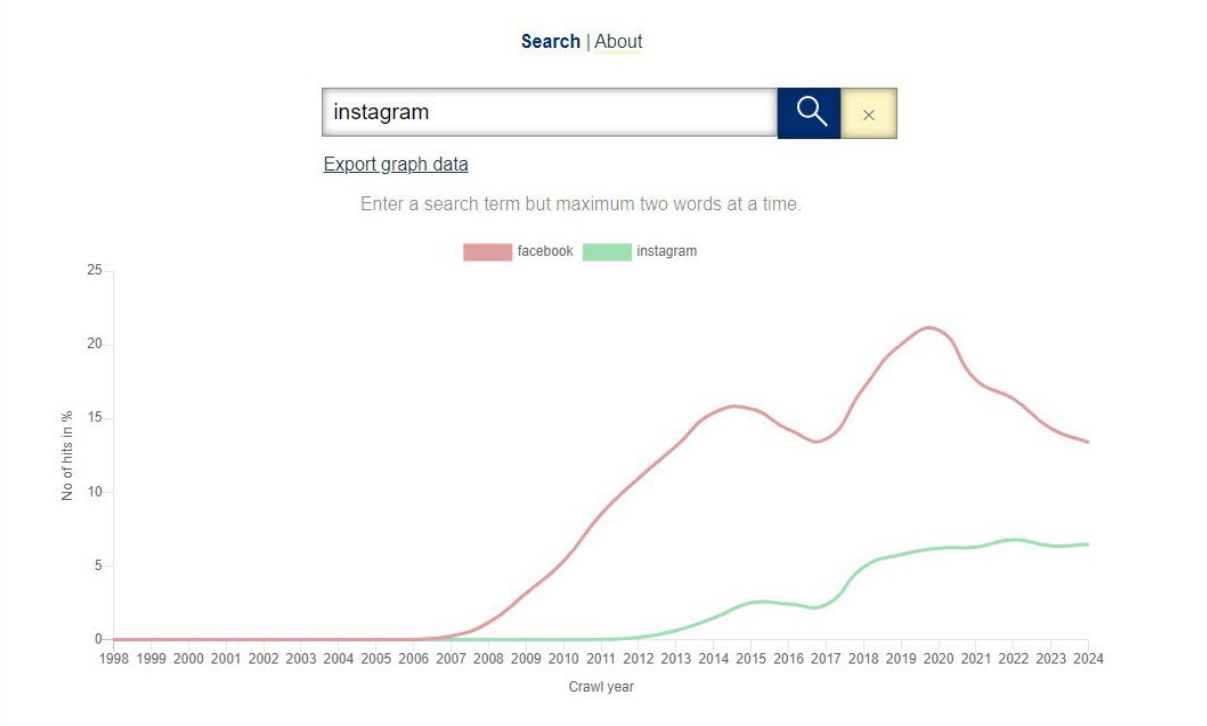


Figure 40: Netarkivet's open N-gram graphs for Facebook and Instagram.

In the example, the N-gram graphs show the frequency of mentions of Facebook or Instagram, respectively. Keywords in this search are entered one by one, and the graph will then be added. Entering both keywords at the same time results in a search for instances where both are mentioned.

The N-gram is interactive; pointing to the curves will open information with exact numbers of instances and percentages per year – but it should be remembered that the instances are counted from archived web pages with unknown numbers of more or less identical versions, and do not represent exact insights into the actual historical web. Please see “Important notice” at the end of this subchapter.

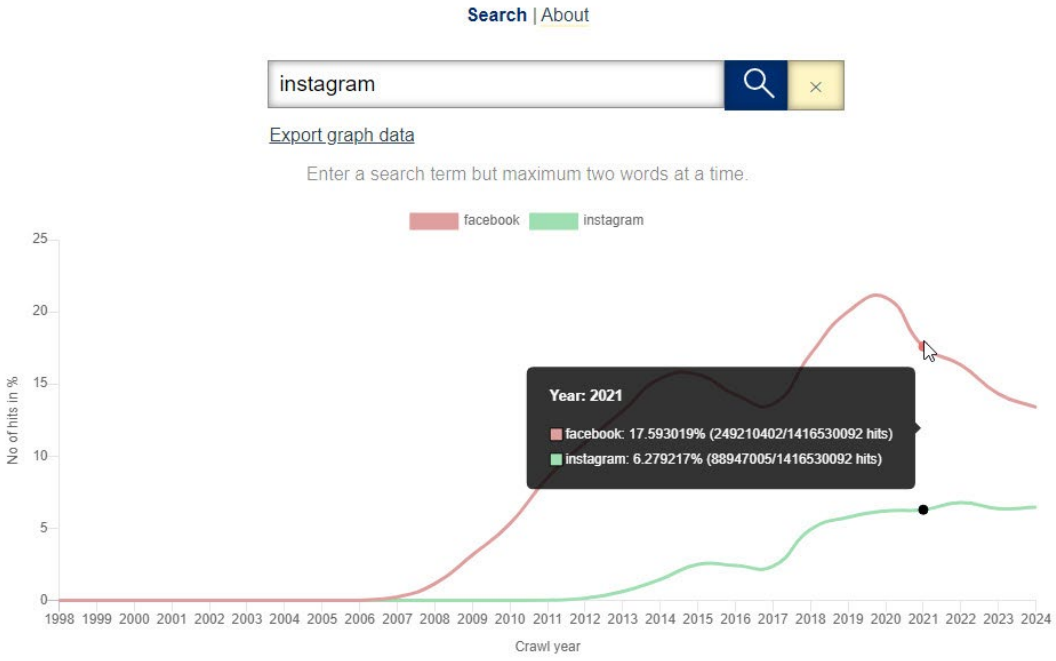


Figure 41: Exact numbers from the N-gram are available on mouseover.

Potential users may get impressions and do basic tests of specific topics with the open functions. If the use and popularity of social media were a research question of interest, then the graphs in the example may lead to new insights, such as the observation that mentions of Facebook have been falling since 2020, while mentions of Instagram have been rising since 2017 but may show a tendency of stabilising at around eight percent. This could inspire research questions such as,

why Facebook, based on the unsorted archive data, seems to be losing mentions, and also whether rising popularity of other social media might be pointed out as a probable cause, or if other probable causes might also be found.

It is important to understand that the analytical tool does not give a precise or fully trustworthy result, since the number of results and the resulting graphs are counted from unknown numbers of copies of different pages found in the archive. Please see “Important notice” in 8.2.5 Examples from Netarkivet, p. 167 for a more detailed explanation of the implications.

8.2.2 Workspace and User Manual

If an application for access is approved for a researcher or a PhD student, a user manual will be sent out by email along with the approval and access information.

The user manual (“Danish: “Netarkivet Brugervejledning”) is in Danish, and provides detailed specifications on how to use and conduct searches in Netarkivet. The user manual can also be accessed when one is logged in to Netarkivet. The functions available in Netarkivet’s interface and the user manual are updated on an ongoing basis.

Access to Netarkivet is obtained via Citrix as a virtual workspace provided for the user in Netarkivet, and tied to the user’s access account. The Citrix software must be installed as specified by Netarkivet.

The Citrix client provides a secure internet connection between the user end and the content and software in Netarkivet. This solution serves as a closed channel: There is no connection to the surrounding internet; only the content in Netarkivet will be found and shown. There is therefore no risk of online interference from third party services on the live Web. See also chapter 7.5 The Risk of Online Leaking, the example in 8.1.4 Examples from The Internet Archive, p. 141-143, and finally the example in 8.2.5 Examples from Netarkivet, p. 164-165.

After logging in to Netarkivet, the user must open the Citrix Firefox browser in order to access the personal workspace with Netarkivet's two user interfaces; SolrWayback and OpenWayback.

This is the Citrix portal just after login:

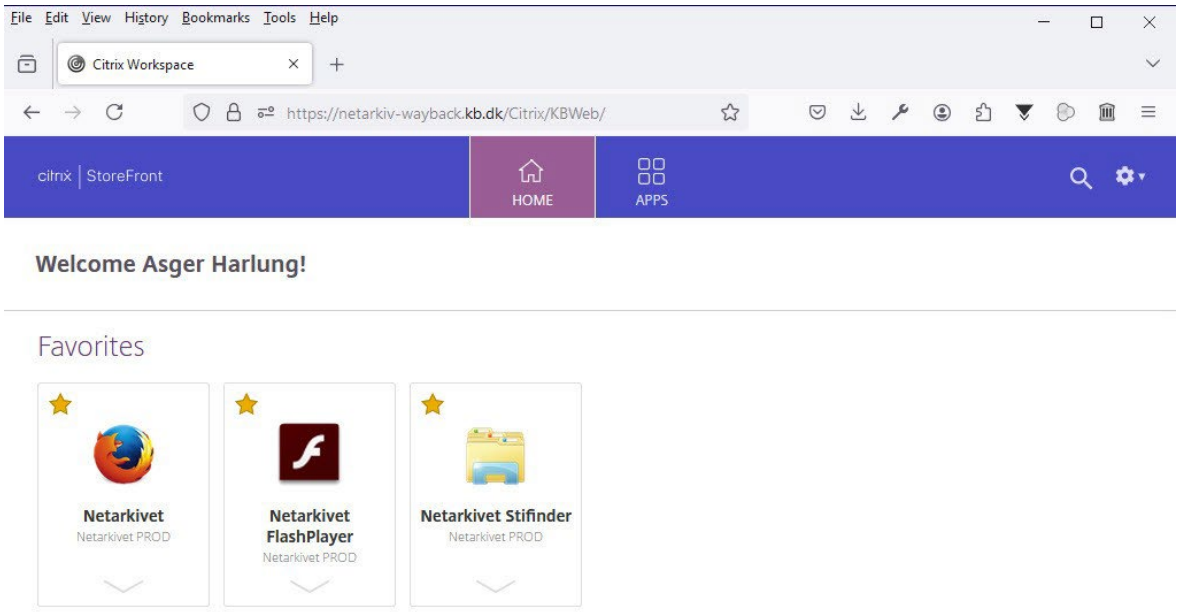


Figure 42: Netarkivet's Citrix portal.

The Citrix Firefox browser, marked with a Firefox logo and "Netarkivet", opens in a new window on the user's computer. It is a fully functional browser, tied to the user's personal workspace as part of the user account. Findings can be bookmarked in the browser, whereby they may be saved and sorted session by session.

The user's workspace also includes a storage space where data extractions can be stored as downloads from the Citrix browser. The storage space is tied to the user's account. Nothing can be downloaded to the user's own computer, but in some cases data extraction can be permitted and arranged for research projects. Information on this along with the relevant application form can be found on Netarkivet's "Research access" page.

In the Citrix browser window the user will be presented with the two web archive interfaces available on Netarkivet. The user manual can

also be accessed from this page (“Brugervejledning”, in the top right corner).

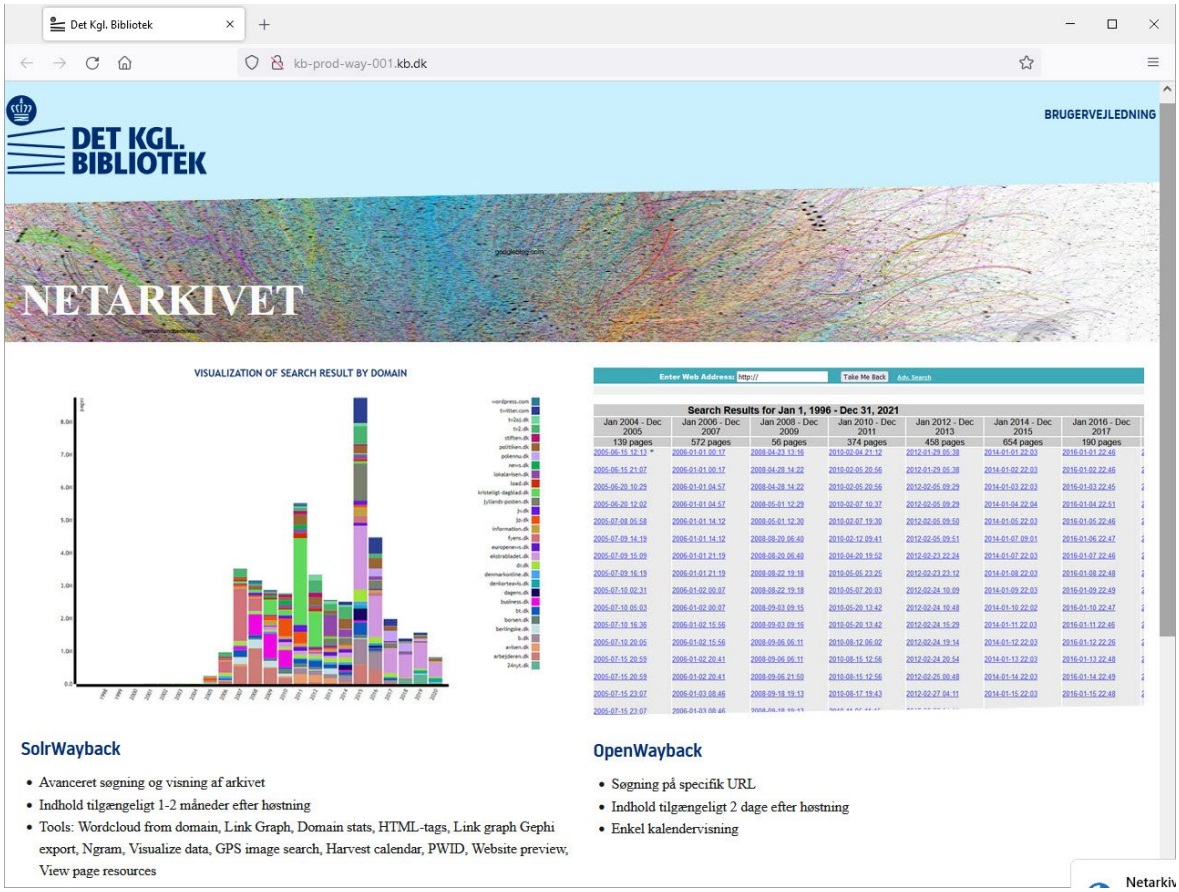


Figure 43: Opening Netarkivet with the two interfaces, SolrWayback and OpenWayback.

The Citrix Firefox browser supports opening new tabs, and thus the two interfaces can be used in parallel. Findings in either interface can be inspected in new tabs, and the user manual can also be kept at hand in this manner.

8.2.3 OpenWayback

OpenWayback is the oldest installation in the archive. It is similar in basic functions and use to the WayBack machine in The Internet Archive. It is planned to be replaced with a newer software, PyWb, wherefore only a short mention is given here.

OpenWayback only supports searches for websites and web pages by URL or domain (HTTP and www prefixes can be omitted). Free text search is not supported.

Due to the basic functionalities in OpenWayback, SolrWayback is the more recommendable interface, which will be used in 8.2.5 Examples from Netarkivet. One example from OpenWayback is given here in order to illustrate its potential of getting an overview of archived copies. A search is conducted for “netlab.dk” with this result:

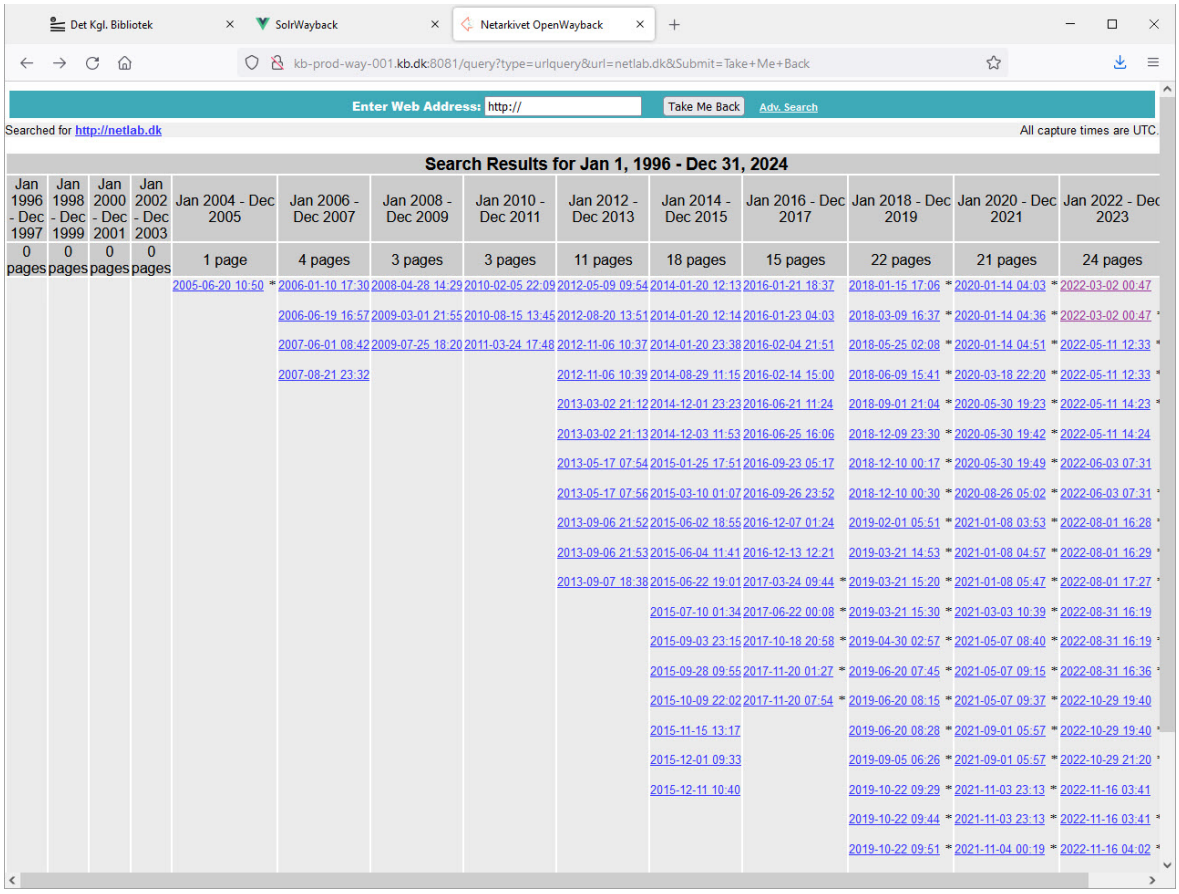


Figure 44: OpenWayback list view for copies of netlab.dk.

This list view can be useful for finding specific copies which may be opened directly or included in work on the other interface, SolrWayback. If opened in OpenWayback the archived pages will be presented with a timeline toolbar like the one in The Internet Archive’s WayBack Machine (see 8.1.4 Examples from The Internet Archive).

8.2.4 SolrWayback

SolrWayback is a frontend build to support research access, discovery and playback.

It was developed at Netarkivet by special consultant Thomas Egense at The Royal Danish Library in 2018, who continues maintenance and development with a team of other developers. SolrWayback is open source software and is also in use at other web archives, presently (2024) at BnF, (Bibliothèque nationale de France), and under consideration for Kulturarw3 (the Swedish national web archive at the National Library of Sweden).

SolrWayback supports advanced search functions, including

- Booleans, such as “AND” as a specification that two or more conditions must be fulfilled in the search results,
- fields which work as various search filters, e.g. timeframe or type of content,
- interactive visualisations where findings may be opened.

SolrWayback also supports data extraction in the CSV and WARC formats. For an explanation of the CSV file format see 9.5.2 API Harvesting; for WARC see 7.7 The Archived Web as Data. However, extracted data will be stored in the user’s Citrix workspace, and can only be obtained for local and more specific use if a data handout is applied for and approved (see 8.2.2 Workspace and User Manual, p. 153).

Finally, SolrWayback supports extraction of PWID addresses. Since Netarkivet is not accessible to the public, this is the most precise way to refer to findings in the archive. See 8.3 Referencing from Web Archives.

PWID addresses are generated for all elements; the main web page, and its parts (i.e., web elements such as images etc.), whereby one

can a) refer directly to the full page or any element, e.g. an image or a document, b) see the harvest time for each element, and c) see all the elements on a web page. This is similar to the “About this capture” function in The Internet Archive described in 8.1.4 Examples from The Internet Archive, p. 141-143.

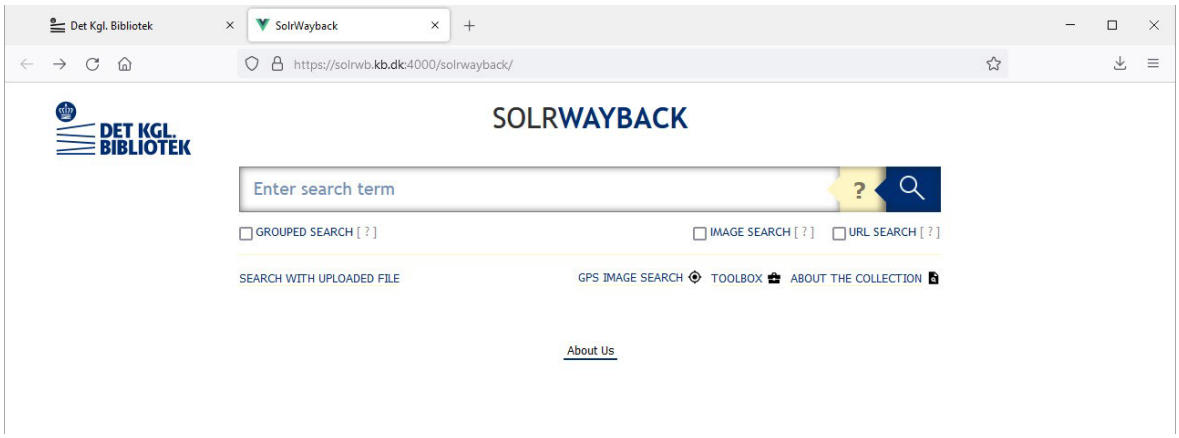


Figure 45: The SolrWayback interface before a search is conducted or a function selected.

Clicking the question mark in the search field will open a help guide with specifications on syntax for conducting advanced searches with fields and boolean commands.

8.2.5 Examples from Netarkivet

Please notice that the purpose of the examples given here is to convey an impression of what one may find in a web archive. Any functions named or shown may change, and new functions will more than likely be added.

In keeping with the examples from The Internet Archive, a search is conducted in Netarkivet's SolrWayback interface for the domain, netlab.dk.

Results are given chronologically, showing the latest first. Results where "netlab.dk" is mentioned are also included. There are a total of 34,242 results. 32,060 of these are from the website netlab.dk itself, and a remainder of 2,182 results is cases where other archived websites have mentioned netlab.dk. The results are for archived copies or versions found in the archive, and no not reflect an actual number of pages that have existed on the live Web.

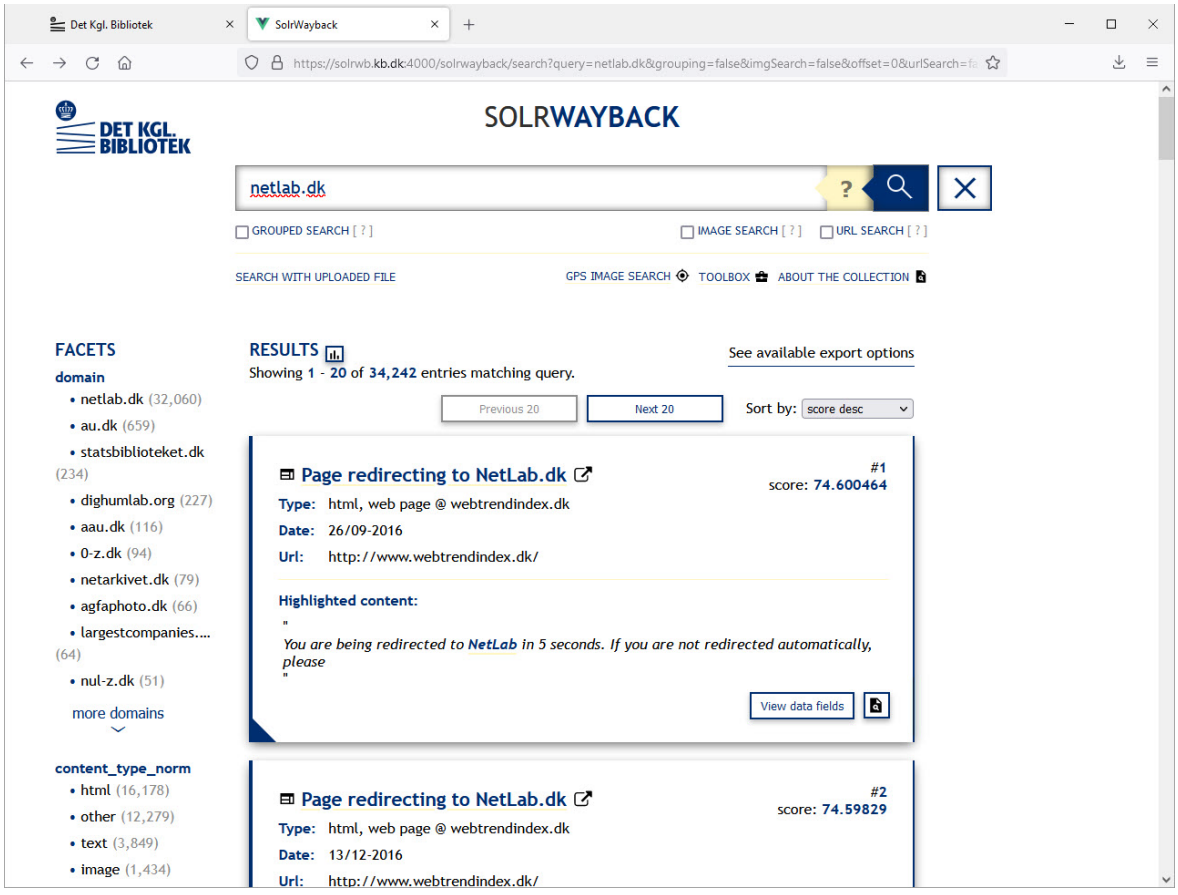


Figure 46: The initial result of a search for netlab.dk

If one wants to see only results from the website source netlab.dk itself, then this is obtained by clicking on that domain in the left-hand list.

This leads to a list of results for that domain only:

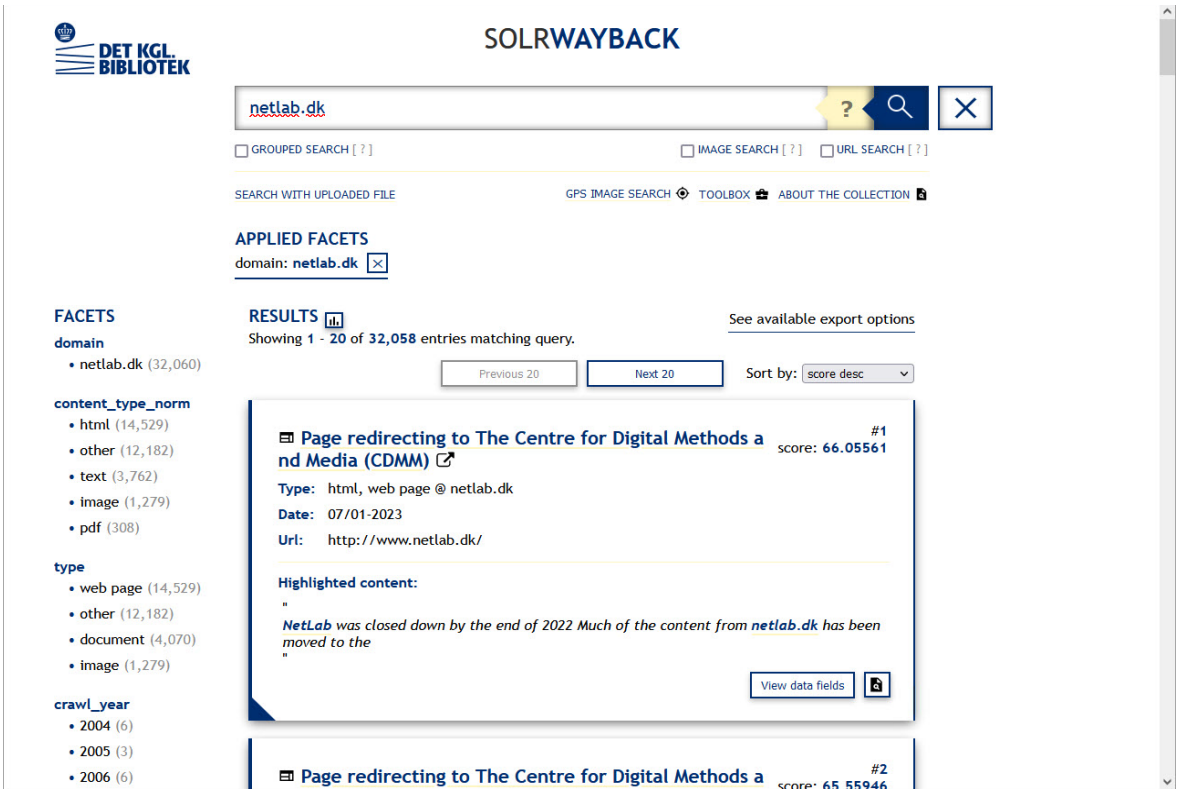


Figure 47: The facet “domain: netlab.dk” limits the search to results from the domain netlab.dk.

In order to go to results for 2022, one can scroll down and select that year under “crawl_year”.

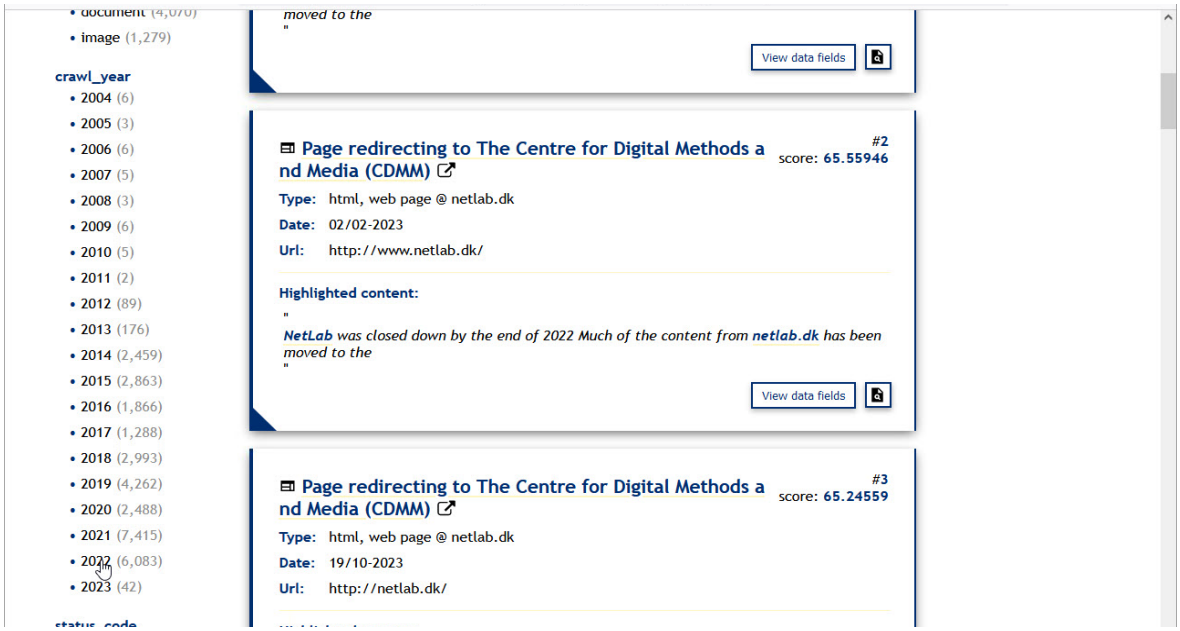


Figure 48: Narrowing results, in this case by the year 2022.

The result of these operations is a list of saves from the domain netlab.dk itself, in the year 2022.

Figure 49: Search results narrowed to harvests of netlab.dk in 2022.

There is more than one way to get to this result. Provided that one knows in advance that a copy of the specific domain from 2022 is of interest, a more direct way would be to do a search with a field specifying the domain combined with a field specifying the crawl year 2022, and a boolean “AND” stating that both conditions must be fulfilled. A search for copies of the domain “netlab.dk” from the year 2022 would be fulfilled with the following search terms in the search line:

“domain:netlab.dk AND crawl_year:2022”

This combined and more direct search term could be generated with the aid of the “help guide”, which is called by clicking the question mark in the search line, and would lead directly to these results. However, users of web archives may often have to narrow down a search step by step as shown in the examples.

On the list of archived copies of netlab.dk in 2022, the first is from March 02, 2022. This is the closest possible copy to the example used in 8.1.4 Examples from The Internet Archive (which was from February 1, 2022), so this will be the copy of choice for a closer look:

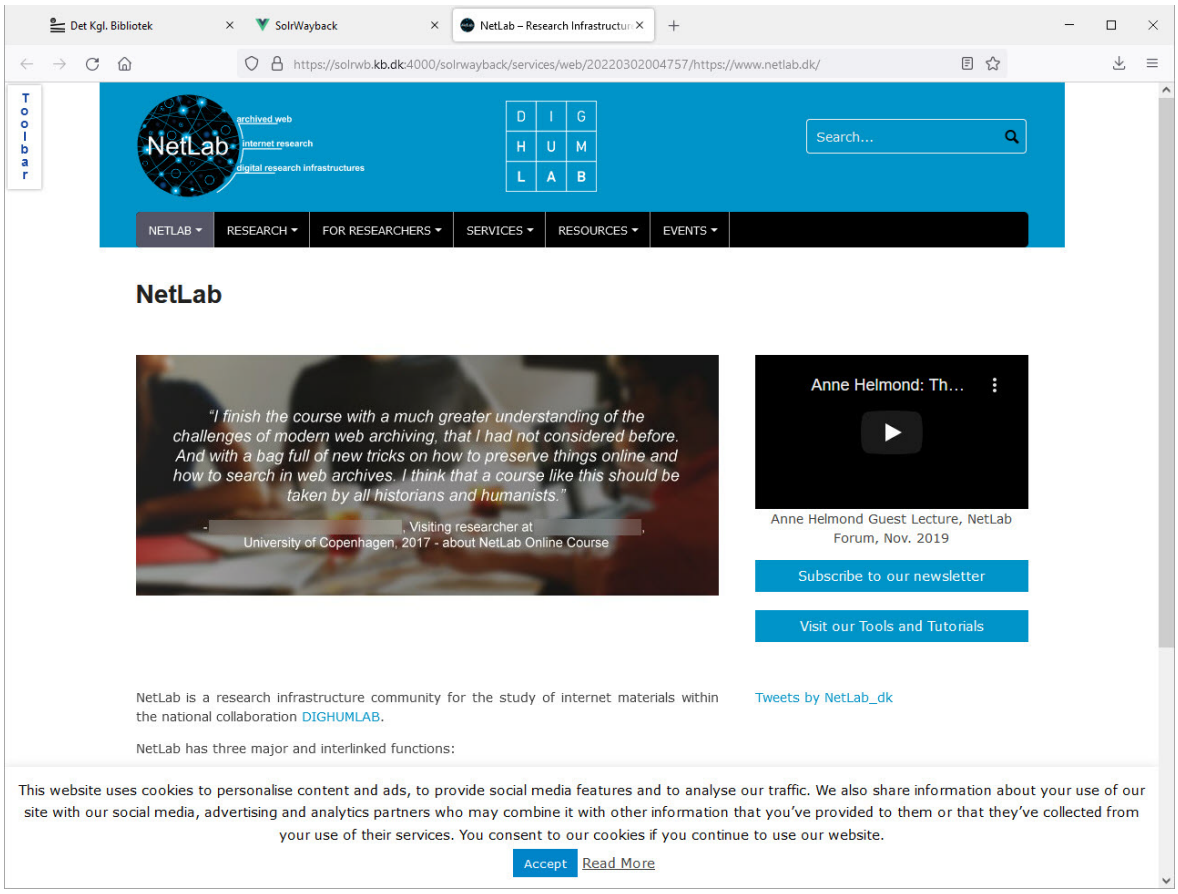


Figure 50: Netlab.dk from March 02, 2022, opened in SolrWayback. Details blurred by the author.

As with the example in 8.1.4 Examples from The Internet Archive, p. 140-141 the page opens with a cookie consent overlay. This can be removed by clicking accept, and no cookies are stored by doing so in the archived copy.

Please notice the video embed for a guest lecture to the right of the banner. Clicking it will result in a black frame, because the video has not been preserved – and because the Citrix Workspace is not connected to the live Web, online leaking cannot occur; the user is working with archived content only. See 7.5 The Risk of Online Leaking.

On the top left corner of the page is an overlay from SolrWayback; a vertical "Toolbar" button. Clicking it and then closing it will remove it completely and one will have to reopen the archived page in order to get it back, while the option of hiding the toolbar will result in going back to having the toolbar button at the ready.

The toolbar provides access to four functions; “Harvest calendar”, “PWID XML”, “Page previews”, and “View page resources”.

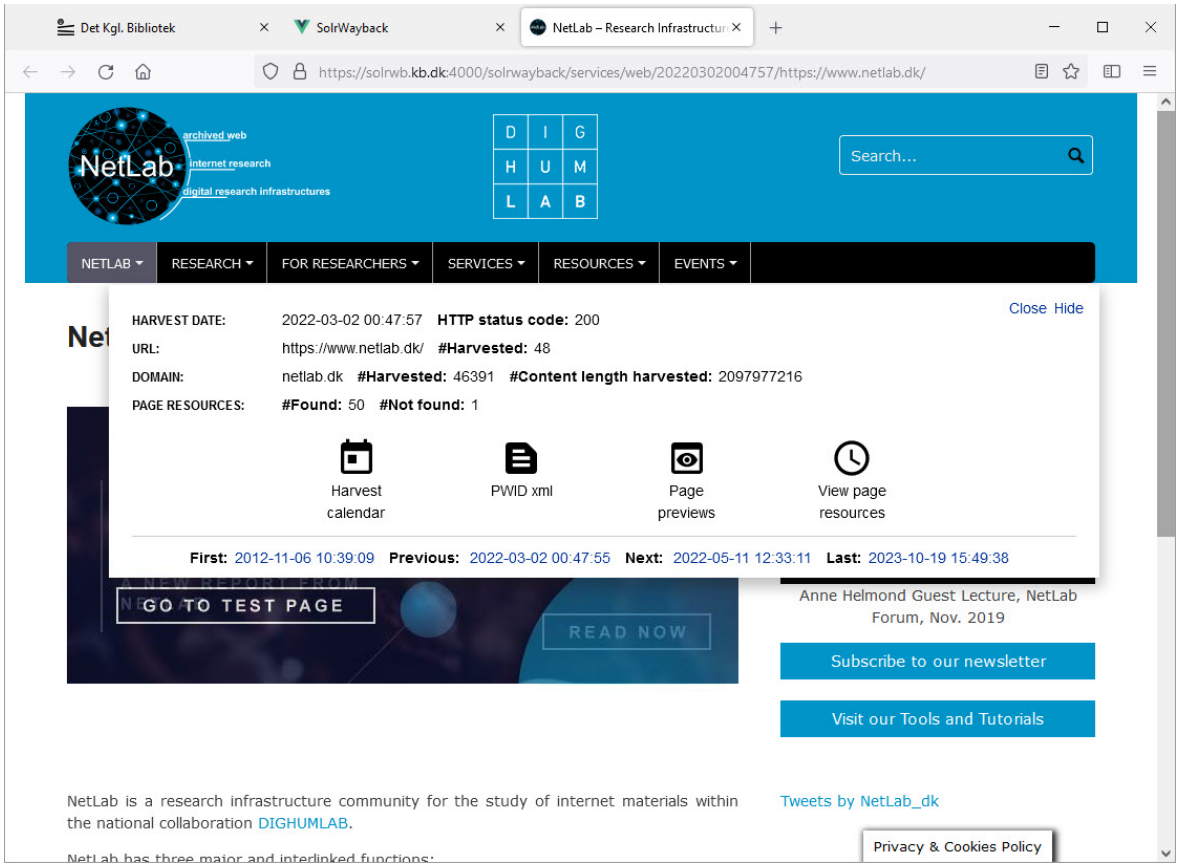


Figure 51: The toolbar with functions for inspecting archived content.

The function “Page previews” has not yet been implemented (in 2024).

The function “PWID XML” generates a list of PWID addresses for all the content presently shown; in this case a full web page and all its elements. The PWID addresses can be copied to the user’s clipboard and saved on the user’s own computer. Readers may notice that the copy of the NetLab Page has an archive URL in the browser’s address line, similar to those seen in 8.1.4 Examples from The Internet Archive, consisting of an archive location, a timestamp, and the original address for the web page. But since Netarkivet is not public, such archive URLs are purely internal and cannot be used as references that will work for others. Therefore the function for PWID addresses provides users of Netarkivet with the best solution for referring to content in a closed archive, or from varying web archives. See also 8.3 Referencing from Web Archives.

“Harvest calendar” opens a graphical overview of copies by year. It is similar to the calendar view found in The Internet Archive where one also finds a graphical representation of when and how many times a page has been archived, by date and month. But the calendar view in SolrWayback differs by showing a full overview for all years:

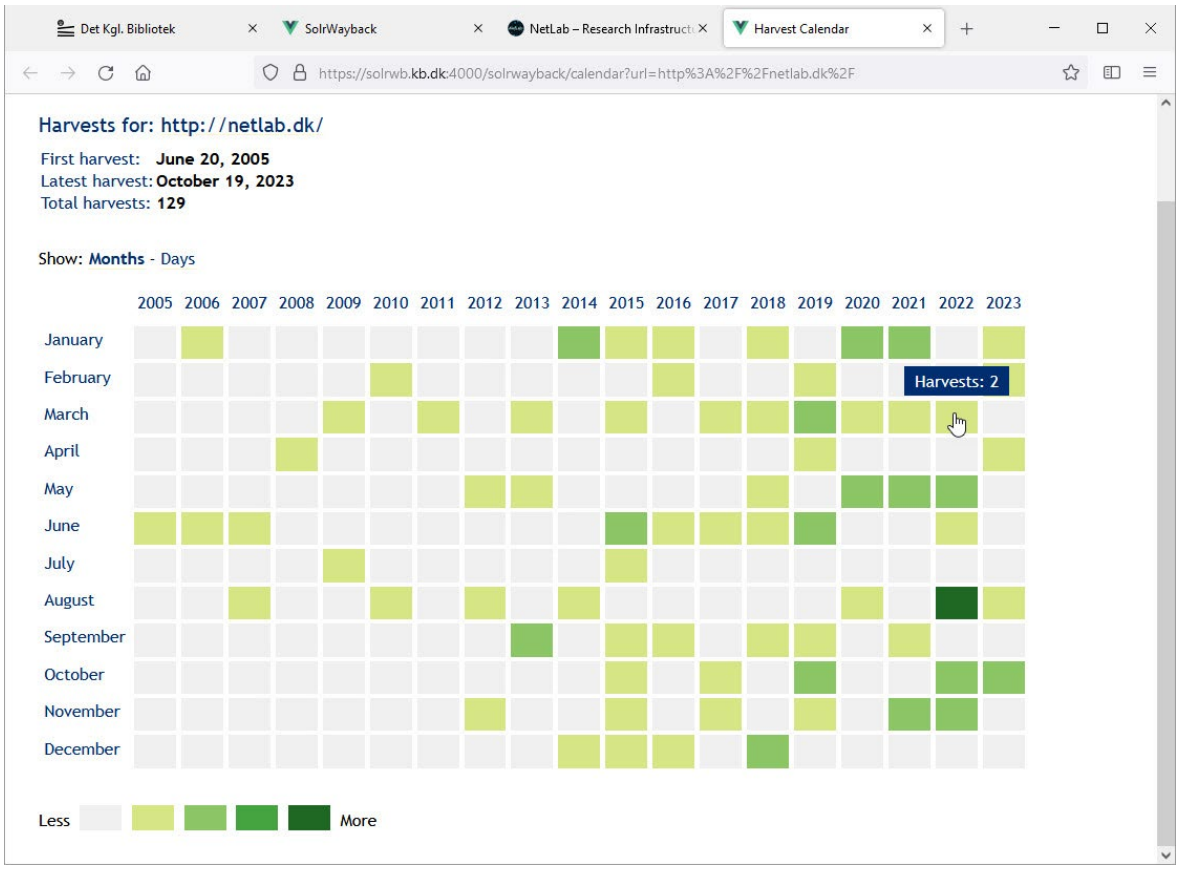


Figure 52: Harvest calendar for netlab.dk. Mouseover reveals that there are two copies for March 2022.

The toolbar function “View page resources” is similar to the function “About this capture” found in The Internet Archive; it provides a list of information on when each element shown was saved.

In the following example the list for the page opened in figure 50 and 51 has been scrolled down, and the YouTube embed from the page has been highlighted. This provides information on the source for the video that would have been accessible on the page when encountered on the live Web in 2022. But Netarkivet has no access to the live Web, wherefore online leaking cannot take place here and no connection to YouTube is established.

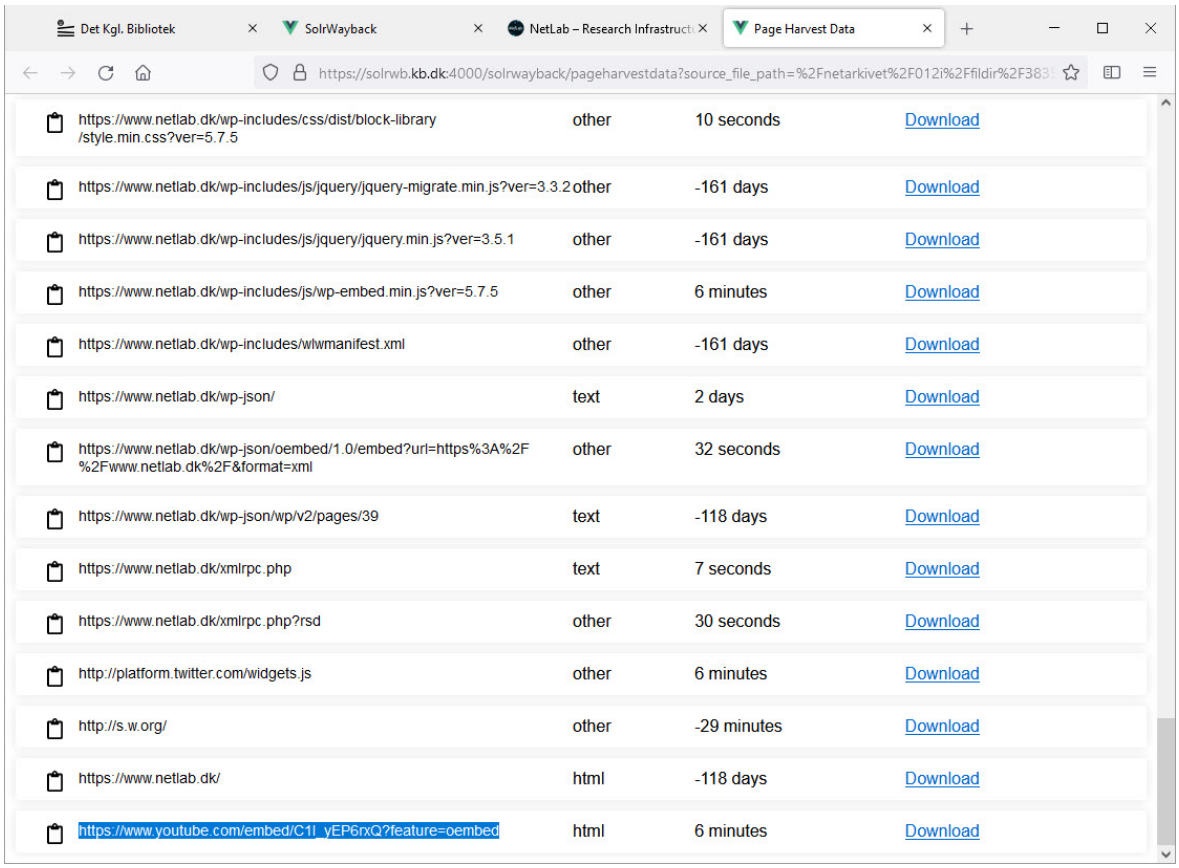


Figure 53: List of page resources with highlighted YouTube embed.

As also seen in 8.1.4 Examples from The Internet Archive, and explained in 7.6.1 Time Jumps in the Content, p. 105-106 there are parts that were harvested before or after the date for the specific copy of the archived web page.

In fact the main page content, the HTML file, was harvested 118 days earlier. The reason that this file has not been harvested on the date for the copy is that no changes have occurred in the HTML file, but other files have been harvested on the date for the archived page in order to preserve a copy as it looked on that specific date.

Besides from finding, using, and possibly inspecting specific copies of web pages, SolrWayback provides other ways to inspect web content, and find information on domains.

Among these are the graphical analyses and representations that can be accessed via SolrWayback's "Toolbox". It can be found under the search field on the SolrWayback entry page:

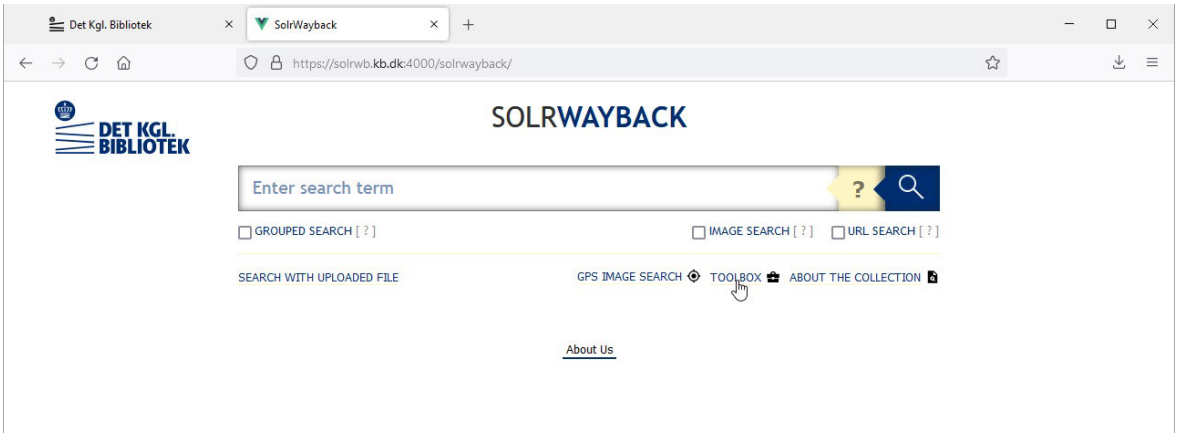


Figure 54: Toolbox is located under the search field.

Selected examples are given by way of illustration, but since the functions are under constant maintenance, updating and development, and new functions are sometimes added, users who get access to Netarkivet should refer to the user manual and the help sections provided in the interface in order to get the best experience and results.

The toolbox opens with the first tool visible, and the rest listed at the top. The first function (presently, 2024) is “Wordcloud”, which can generate a wordcloud for a domain. No advanced settings are presently available; one cannot select a timeframe or a year, define stopwords that should not be included on the wordcloud, e.g. “cookies”, or edit the colour palette, text direction, or similar.

Without a means to sort the data this function draws upon all archived copies of pages from a given website, so the results may not reflect a website accurately as it would be from a cleaned dataset with only one copy of each page from a specific timeframe.

Important notice:
The analytical tools in Netarkivet, including the public version of N-gram described in 8.2.1 Access and Appetisers, all draw directly on data from the archive with unknown numbers of copies and versions of different pages. The tools thus allow insights into counts from the archive, but this does not reflect historical observations accurately. Any possible trend showing in the analytical tools would therefore have to be verified from a cleaned dataset with versions removed. The archive supports creation of researcher's own findings in a personal storage, but the analytical tools described in this chapter are not presently capable of treating the user's own and cleaned dataset(s).

Any web page may exist in any number of copies due to by-harvests (see 5 The Web Archiving Process p. 73-77) as well as different harvest strategies for different pages types (see 8.2 Netarkivet, p. 148-149). Some pages may have very many versions, while others may only have a few. A trend that exists when counting instances of archived copies may thus be rather different from what would be the case with a cleaned dataset with representative versions selected and duplicates removed in order to more accurately reflect the individual websites for a given timeframe. Therefore, in order to verify a trend, if possible, and in order to establish a historically accurate analysis, researchers will need to have their relevant data extracted for data cleaning, followed by their own analyses with similar tools. Information on data extraction from Netarkivet is given in 8.2.2 Workspace and User Manual, p. 153.

In this example, a wordcloud for netlab.dk is requested:

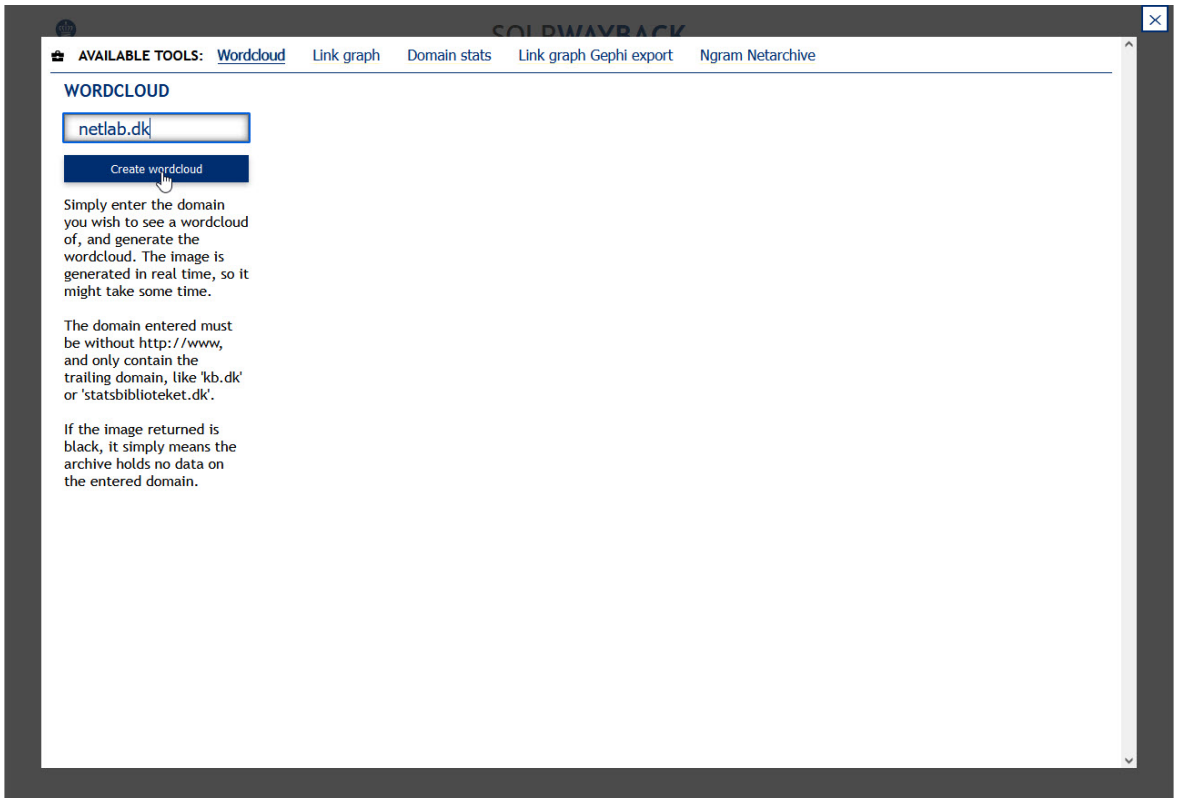


Figure 55: Requesting a wordcloud for netlab.dk.

And this is the result:

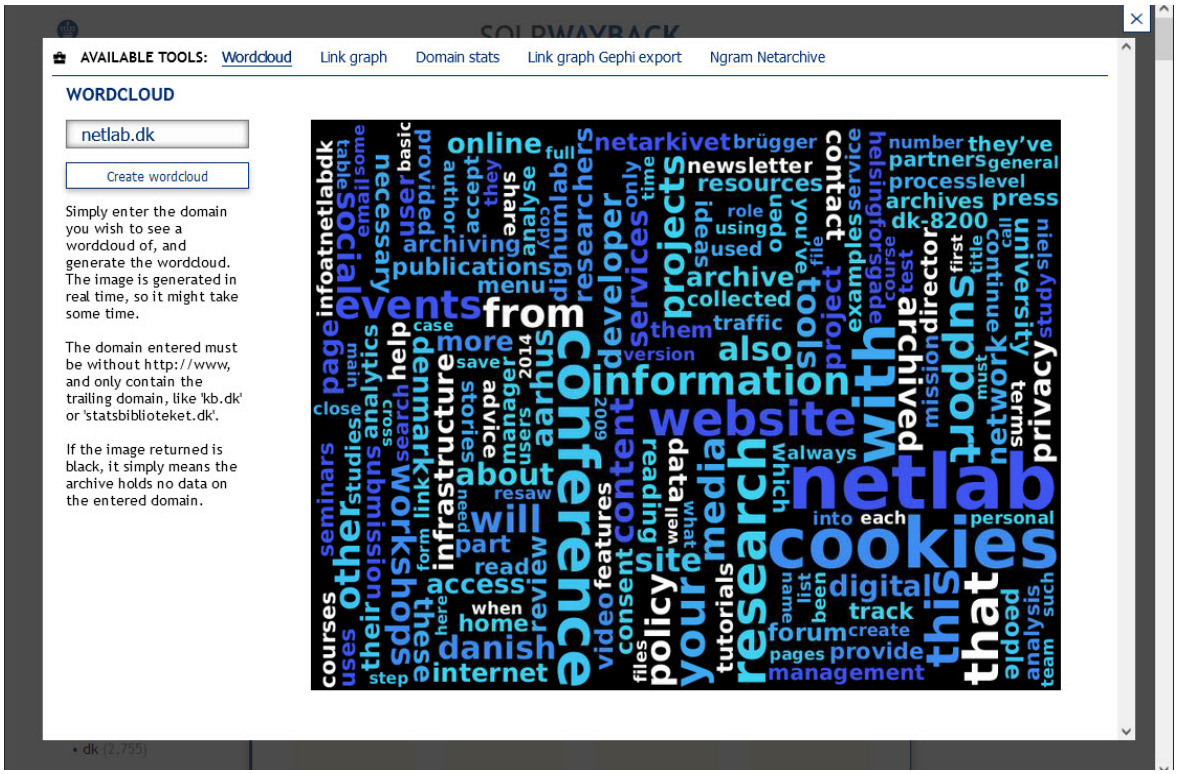


Figure 56: Worldcloud generated for all copies of netlab.dk.

Keeping in mind that the results are extracted from raw archive data with unknown numbers of duplicates, the wordcloud function may help a researcher to notice keywords worthy of further inspection for a domain.

Another function is “Ngram Netarchive”, which is the same function as provided on the open main page for Netarkivet; (see 8.2.1 Access and Appetisers), but the internal function when found in the archive supports timeframes and more queries.

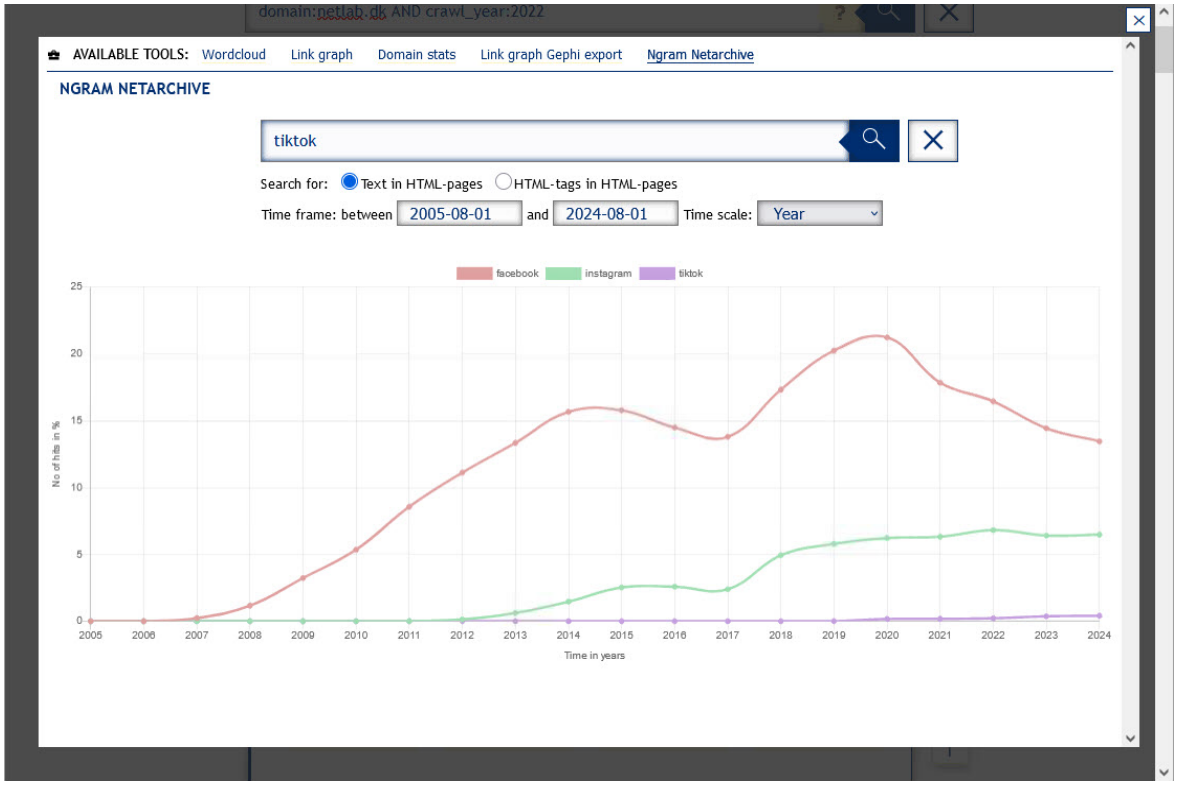


Figure 57: N-gram for Facebook, Instagram, and TikTok.

In this example the searches for Facebook and Instagram that were also conducted in the open version have been repeated, but in a time frame starting in 2005. Another search has been added for TikTok.

The results suggest that the seemingly rising popularity of TikTok from 2019 and forward would not suffice to explain a corresponding decline for Facebook after 2019. But these results are at best a suggestion of possible trends since they still come with the drawback, that they are based on all versions of pages stored in the entire archive in the requested timeframe and therefore cannot accurately reflect what

occurred on the live Web. Please see the discussion in "Important notice", 8.2.5 Examples from Netarkivet, p. 167.

Possibly, the most interesting interactive visualisation is “Link graph”, which can provide a network analysis for a domain. Network theory is an academic discipline in itself, but popularly speaking the Link graph can tell the user how a website is connected to other websites.

Ingoing link direction will show other websites that referred to the target website; outlinks will show other websites that it pointed to.

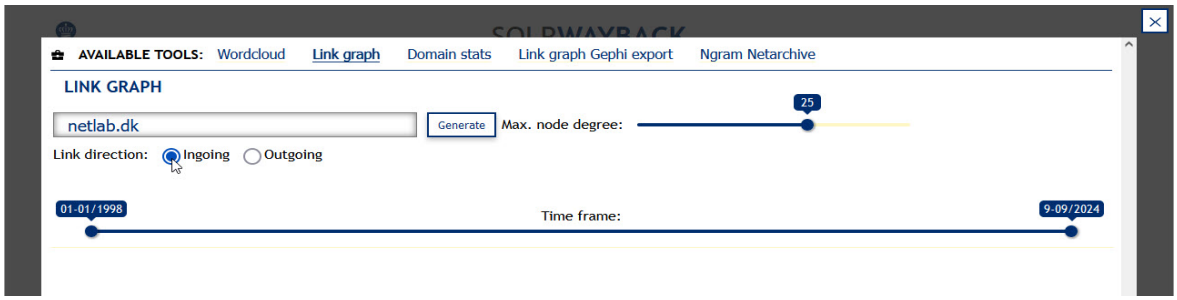


Figure 58: The Link graph function with its default settings.

The default settings will result in a major network mapping including not only websites that pointed to the target website, but also the network of those websites, since people might find website X by following a reference from website Y to website Z, and then decide to follow a reference from website Z to website X, etc.

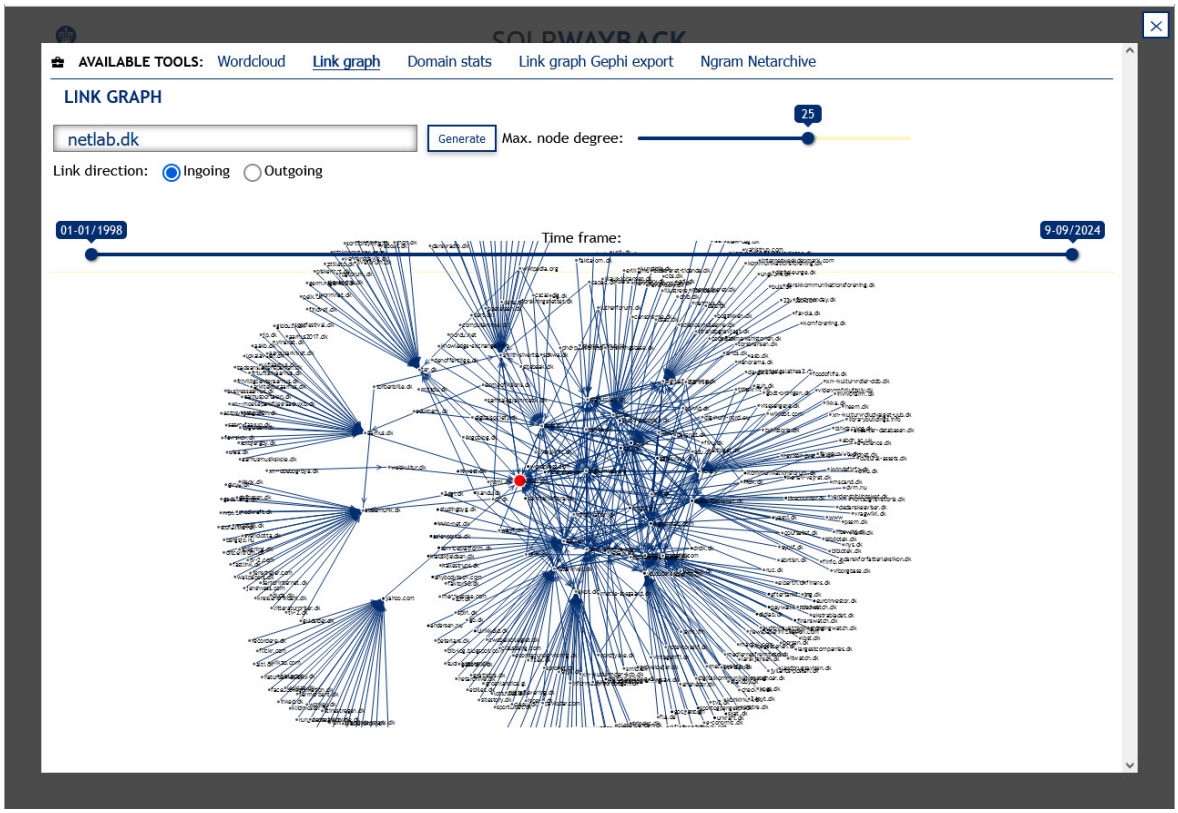


Figure 59: network for netlab.dk, default settings.

A more delimited search can be obtained by decreasing the time frame for the search, and by lowering “Max. node degree”. The latter will “zoom in” on a network that is closer to the target website.

Once again it should be noted that this function works with the raw data from all the existing copies of pages in the archive. If one website has hundreds of copies it will count as a closer connection to the starting website than a single page with just two copies from another website which was actually closer in the network on the live Web in the same timeframe. The Link Graph in Netarkivet will thus show websites that were actually part of a website’s network in a given timeframe, and the observations may be worthy of closer inspection, but the results will not be accurate. Please see "Important notice", 8.2.5 Examples from Netarkivet, p. 167 for more details.

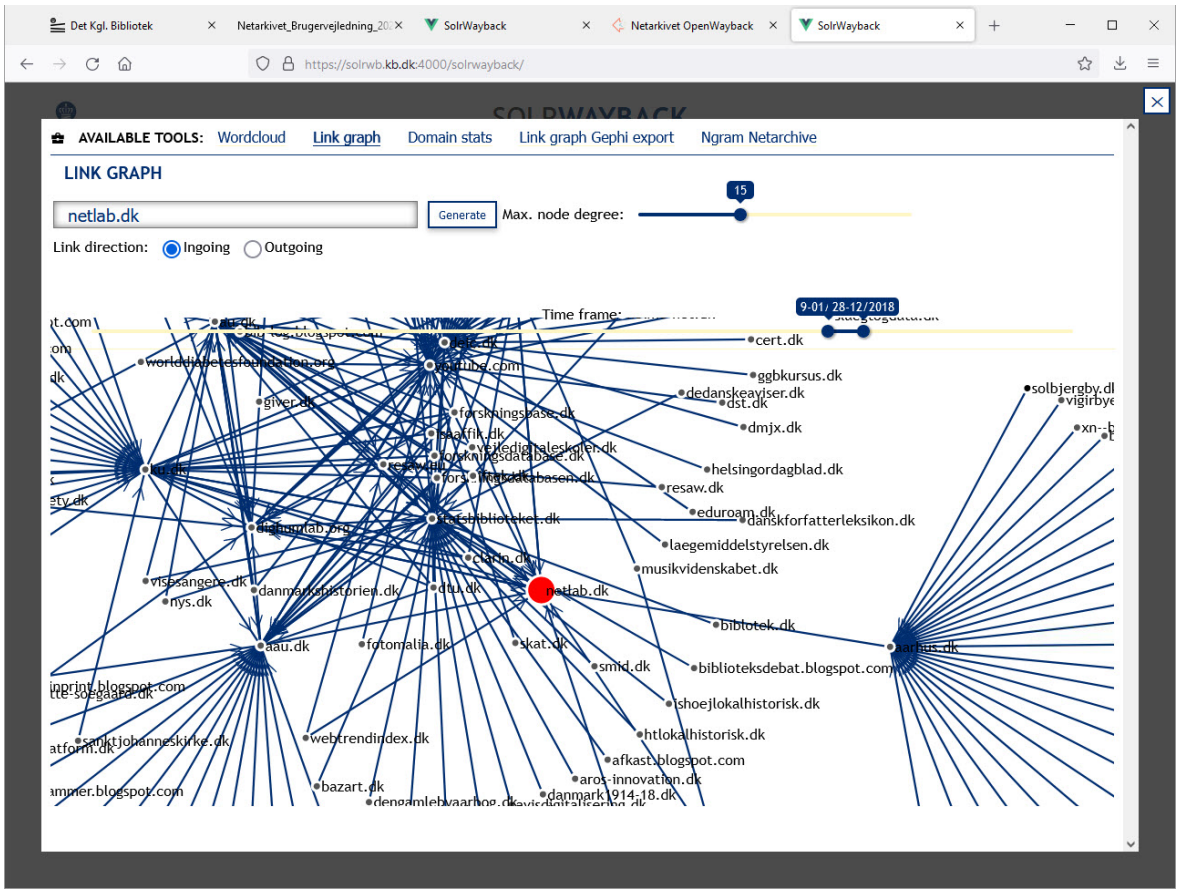


Figure 60: Slightly more delimited network with timeframe and max 15 nodes.

Websites that refer to the website one has searched for are also found and listed in a standard search in SolrWayback, as shown in the first example in this chapter (figure 46).

8.3 Referencing from Web Archives

When a web page is referenced as a source in an academic text, the widespread and formally established tradition is to give author name (or name of the entity with authorship), title, the URL for the page in question, and a retrieval date where the cited text was found online.

However, given that most online resources are likely to become irretrievable within a few years (see 1 The Importance of Web Archiving, p. 17-18), it is a much better strategy to refer to an archived and publically available copy of a web resource. This will serve as a more stable reference, where the cited content can almost invariably be found and confirmed.

If a page that one plans to cite is found online, then saving it to an open archive such as The Internet Archive is the quickest, safest, and most accessible way to ensure that one has a useful and stable reference in the form that one found it. The way of doing this, with the limitation that some pages cannot be saved for technical reasons, is described in chapter 8.1.2 The 'Save Page Now' Service.

As explained in 8.1.3 Internet Archive URLs an archive URL from The Internet Archive contains the address of the archive, a precise UTC time stamp for when the copy was archived, and the original address for the web page in question.

A common reference style could thus consist of:

- Name of entity/entities with authorship, e.g. personal name, organisation name, name of website,
- The title of the web page, e.g. article name,
- URL to relevant copy in The Internet Archive.

The challenge with this reference style is that it may not resolve in cases where archives change their web archive URL pattern. This can happen either because a web archive changes their access tool, content location, or because they change their domain. The latter happened for the web archive at the National Library of Ireland when they changed service provider for their web archive (Aturban et al. 2021).

In cases where content has been found in an archive with restricted access, it can be worth seeing if a suitable copy from approximately the same time and with the same content can be found at The Internet Archive or another open archive (see 8.4 Other Web Archives).

If a page cannot be archived in The Internet Archive or found by trying in another archive, then the traditional form of reference mentioned at

the start (URL + retrieval date) may be necessary. In such cases it is advised to make a local copy as documentation (see 9.2 Basic Archiving, Single pages).

If a page was found in an archive with restricted access, and a copy in an open archive cannot be found or created, then the best option is to use a PWID URN address. PWID URN addresses should also be considered if referencing resources from different archives.

PWID stands for “Persistent Web Identifier”, and is a standard for web archive referencing developed by Eld Zierau at Netarkivet, and URN stands for “Uniform Resource Name”. PWID URNs consist of a specification of the reference type as an URN in the PWID format, “urn:pwid”, followed by four identifiers consisting of:

1) Archive where content was retrieved,
2) Date and time of archiving,
3) Specification of either "page" for full web page, or "part" for page element,
4) The archived URL (the original URL for the archived content).

All specifying elements are separated by colons.

A PWID URN identifier for a resource used in this article;
Alexis Rossi (2017): If You See Something, Save Something – 6 Ways to Save Pages In the Wayback Machine; Internet Archive Blogs,
— is as follows:

urn:pwid:archive.org:2024-07-31-T13:51:22Z:page:https://blog.archive.org/2017/01/25/see-something-save-something/[29]

This PWID URN is a detailed transcription of the corresponding direct Internet Archive URL,

[29] Notice that the timestamp in a PWID URN is separated by year, month, date and time of day as opposed to an Internet Archive URL where the numbers are given as one string.

https://web.archive.org/web/20240731135122/https://blog.archive.org/2017/01/25/see-something-save-something/

Users of Netarkivet's SolrWayback interface can automatically generate a list of PWID URNs for any resource visited and use these as direct references to pages or their separate web elements.

For a detailed presentation of the design and rationale for PWID URNs please refer to "URN Namespace Registration for Persistent Web IDentifiers (PWID)", or the interactive PWID Poster which also offers advice on use cases for different ways of referencing archived content, both by Eld Zierau (2022) in References.

8.4 Other Web Archives

There are many web archiving initiatives around the world. To some readers a national web archive may be the most relevant to their research, for others open archives such as The Internet Archive may be an ideal place to start.

As mentioned in the chapters 7.3 Missing Pages, and 7.4 Missing Content Elements it is a good idea to try other archives if specific content cannot be found in the archive where one first tried.

The Internet Archive, the Portuguese (https://Arquivo.pt/) or Icelandic (https://vefsafn.is/) web archives are examples of institutional web archives that are open to the public.

There are two recommendable ways to look for relevant or alternative archives:

1) Visit the website for the International Internet Preservation Consortium (IIPC) and search for members or projects. Most large-scale archiving initiatives are represented in the IIPC. The URL for IIPC is https://netpreserve.org.

2) Visit Wikipedia's "List of Web Archiving Initiatives". The list is originally based on a survey published by the Portuguese Web archive, Arquivo.pt in 2011, and is still being maintained in 2024. The URL is: https://en.wikipedia.org/wiki/List_of_Web_archiving_initiatives.

8.4.1 Curated Thematic Collections

Some readers may also have an interest in collections that are curated by archivists, and offer quality content on specific topics. There are at least two places to explore for such thematic collections:

Library of Congress Web Archives
(https://www.loc.gov/web-archives/collections/), also mentioned in the opening of this main chapter, provides curated collections for a range of topics such as events or international politics.

IIPC Collaborative Collections
(https://netpreserve.org/projects/collaborative-collections/)
Members of the IIPC collaborate to create thematic collections of broad interest, including research interests. The collaborative collections are open to the public, and include themes such as, Street Art, War in Ukraine, Novel Coronavirus (Covid-19) (2020-2023), Climate change (2019), or Artificial Intelligence (2019).

The URL for the collaborative collections is presumably stable, since it is a web archiving initiative intended for public accessibility and moving it would be counterproductive to such a purpose. If the URL should move, change or disappear, one may search the IIPC's website (https://netpreserve.org) for information on it.

9 Making Your Own Archive

Takeaways

♦ Provided that it still exists, readers are recommended to check the list of software and services in the Data Collection section of CDMM's Tools and Tutorials list; referred to in 10 Searching for Software and Services, p. 226.

♦ When harvesting data yourself, legal and ethical frameworks must be observed.

♦ Your own local archive is completely designed and controlled by you, with your own choices of depth and width.

♦ Document what you do. Keep and remember to maintain a log. This will be important for you if you need to go back and make adjustments, and also for describing your methods.

♦ Consider saving content in different formats that can supplement one another, and may have different uses.

♦ You may need to have – and sometimes actively use – administrator rights on your computer in order for some programs and services to be available and work properly.

Setting out to do one's own archiving can have many reasons and start out in many different ways. It may start without any actual research plan, by simply saving content of interest on the notion that it would be nice to keep it. It may start with a debate or a new trend appearing online, on the notion that the more of this can be documented and studied, the better. Or it may start from other reasons or interests.

The biggest differences between doing one's own web archiving, and using existing web archives are that,

- As one's own web archivist one controls the exact topical framework,

- One (in combination with one's resources in the form of software, bandwidth, storage space, and IT skills) decides the delimitations; how deep, how broad, how many, and how and if media are combined, e.g. by combining harvests of commentaries for videos with the videos themselves,

- One will be their own curator, with the options of cleaning and sorting data so they are suitable to one's research purposes.

Another big difference is that the existing web archives have content up until and including today, but they do not yet have content onwards from today. In order to look at older content one must use existing archives, but in order to make sure that data from the present and onwards is stored, the initiative of starting to archive it immediately will be a very good idea.

Some research projects will need to look back at earlier content as well as content from the present and onwards. Using existing archives for the former and one's own data collection for the latter will then be a meaningful approach.

Finally, doing one's own archiving is always a good idea for projects that are not heavily data demanding. A research project on, e.g. the

rhetoric and arguments of politicians as they proceed in a newly started debate, may demand only that the researcher has a good collection of articles which may be easily preserved on an ongoing basis.

Small scale archiving can easily have big research potential, wherefore it is a good idea to consider to simply get into a routine of saving any content that one thinks might be useful on a later date.

For most types of web content archiving one will need relevant software or services for the type(s) of content that one needs to archive.

The chapter 10 Searching for Software and Services will attempt to advise on that. Of special note, please refer to the beginning of that subchapter for a reference to the Tools and Tutorials section at Centre for Digital Methods and Media.

Some basic advice is in order before starting out on one's own archiving:

9.1.1 Data Responsibility

First and foremost, when harvesting data from the Web, one (or in some cases an IT administrator at one's institution) becomes responsible for the data.

There are rules and concerns to observe, and as data responsible one is first and foremost bound by a legal framework; rules for copyright, sensitive data, etc.

The rules differ from country to country, and policies and guidelines differ from institution to institution. It also matters whether one is a researcher or a student; the rules to observe will be different in most cases. A broader treatment of this is offered in chapter 11 Legal and Ethical Concerns.

9.1.2 Administrative Rights on the Computer

Most software and services for saving various types of web data are ready to install or use without any special preparations, and they very often come with default settings that will satisfy most users' needs. In other words, one will often need to do little except from starting a program or a service and go directly to using it for the type of content saving that it is meant for.

However, there are a few concerns which become more relevant the more specialised a program or service may be.

First and foremost, if one is using a work computer from one's organisation it is most likely set up and administrated centrally by an IT department.

This can sometimes mean that a work computer is blocked from installing third party software which is not specifically approved by one's organisation. If that is the case, one should request administrative rights for one's work computer. If this is declined, a computer that is not controlled by one's organisation may be necessary in order to proceed. However, the argument that one needs to be able to install software in order to actually do one's job should be a convincing one.

But also, some of the programs that one might need for more specialised purposes may need to access protected parts of the operating system.

For this reason; when installing a program, one may consider installing it with administrator rights rather than just installing it normally. How to do this depends on one's computer and its operative system, and is likely to change from time to time, wherefore one will need to search for or ask for the specific way to do this on one's present operative system.

Active use of administrator rights can wait until encountering a program that will not install or work correctly, but installing with

administrator rights as a standard procedure can spare one from first installing a program, then finding that something is off, uninstalling again, and reinstalling with administrator rights.

Once a program is installed, it may also need to be started with administrator rights. It can be set to do so every time one starts it. Again, how to do this is something that may vary, wherefore once again one will need to find out how to do it either by asking or searching for the correct procedure.

Please notice that starting a program with administrator rights may result in security prompts asking, “Do you really want to do this?”

Getting security prompts every time one starts a program can be an annoyance, so in the end it is an individual strategic decision whether to a) always install and run programs with administrator rights, or b) install programs normally by simply running their installers, and only use administrator rights when absolutely necessary.

9.1.3 Make a Log of What You Save and How

In any type of research, one will need to describe one’s methods so that:

a) One has a solid understanding of one’s data, where they came from, how they were obtained, and what research purposes they were meant for,

b) If one should find that their data is flawed, incomplete, or insufficient in respect of one’s project or analysis, one will have a clear understanding of what one already has, where to go to find the missing parts, and an idea of what will be needed to get them,

And finally,

c) So that others can understand what one did and retrace one’s steps if necessary.

Maintaining a log can be done in many ways, e.g. in a simple text file or in a spreadsheet.

It is advised to make notes on:
- where the data was found,
- the date of saving it,
- the service or application used to save it,
- the settings used (if applicable),
- tags indicating what the data was saved for, e.g. topical cue words.

Creating such a log on an ongoing basis may not be something that one is used to doing, and it is easy to forget adding notes whenever something new is saved to a data collection. But it will be helpful when describing methods, for structuring data for analysis, and not least, for correcting errors or unforeseen needs if necessary.

If one has written academic texts or reports one has also most likely experienced the importance of building the reference list on an ongoing basis. This is much easier than going back and retracing quotes and references for creating a reference list at the end of the writing process.

Maintaining a log of one's data collection is at least as important, and may prove to be more so if one should need to go back in order to make adjustments.

In the process of archiving, one should consider saving the same data in more formats. See 9.2 Basic Archiving, Single pages for examples on how various formats for saving single web pages can have different uses.

9.1.4 Remember to Check the Results

Always perform a quality check. If a program runs as it should and finishes successfully, it may be tempting to assume that everything is now in order, and that the data that one needs is now preserved.

But with web content, its transience and constantly developing technologies, one can never be sure that the results are fine, that they are complete or simply “look good”, without actually opening the saved data and verifying that the result is useful.

Therefore, always remember to open a saved version, and verify that the content has been captured, in a satisfyingly complete and useful format.

9.1.5 It Is Worth Having More than One Browser

Browsers treat and handle web pages differently. While a web page may at first glance look identical when viewed in different browsers, there may be subtle differences – and sometimes less subtle, because specific functions or services may work in browser X and not in browser Y.

Even before one considers web archiving, it is thus a good idea to have more than one browser, simply so that one may use an alternative to the standard browser if something does not work satisfactorily.

But different browsers also come with different analytical tools for users who might want to look at the underlying code (see 4.1.3 HTML: HyperText Markup Language, p. 58-62).

Last but not least, browser add-ons can be very helpful for archiving purposes. Add-ons, in some browsers called “extensions”, are available for most modern browsers from built-in catalogues that can be searched by name or function. Most add-ons are free tools, often developed in communities that are dedicated to a specific browser. Thus, an add-on with a specific function may be available for one browser but not for another.

Both on Windows and Mac OS the standard browsers included with the systems (presently in 2024, Edge in Windows, and Safari in Mac OS) have options for adding add-ons, but for archiving purposes it is

strongly recommended to also have the browsers Google Chrome and Mozilla Firefox at hand.

Different browsers can be used to test different attempts at saving web pages in various forms, and the more one goes into web archiving and content analysis, the more useful one will find having more options.

9.2 Basic Archiving, Single Pages

The most basic ways of archiving are likely to remain relevant. That is, the principles and functions are likely to remain the same, while the functions may change names, specifications, locations, and relevant applications.

These basic forms are the ways to save or capture content directly when it is encountered. They consist of:

- Saving web pages directly to the computer (three primary ways)
- Taking screenshots, or screen recording
- Saving to documents (three primary ways)
- Archiving a page at The Internet Archive (or other open archive).

9.2.1 Saving Directly from the Browser

The most easy and direct way to save a single web page is by pressing your system's keyboard shortcut for saving (ctrl+s on Windows, cmd+s on MacOS) and selecting a file format and a location to save to. It should be noted that saving pages does not work well for web pages that are heavily scripted (see 5.1 Challenges for Web Crawlers, p. 79), or where the primary content is controlled by an API (see 6.1 APIs and API Access).

When testing the result by opening a saved page in a browser, one may find that:

a) An online call from the saved web page can result in a cookie consent pop-up which keeps returning, possibly obscuring the entire page.

b) Using different saving options gives varying results.

Direct browser saves are in other words rarely flawless but may suffice, or be combined with other types of copies.

Choosing “save entire web page” may give a result where embedded content will appear from the live web, and where an HTM or HTML file (HTM or HTML are different system names for the same file type, and are used interchangeably) is saved alongside with a folder; e.g. a file named “Centre for Digital Media and Methods.htm” with a companion folder named “Centre for Digital Media and Methods_files”.

One good thing about this is that the files that were integrated directly on the web page from the same website that it was found on will be saved in the companion folder. Therefore this saving method is useful if one wants a separate folder with pictures and other files from the web page. But it is less useful if the saved page is disturbed by pop-ups or other kinds of overlays.

In the following example of a direct save with crtl+s (in Windows), and choosing “complete web page”, the address in the browser address bar shows that this is a local copy in the user’s own folder system being tested. The URL visible at the bottom discloses that a slideshow is being called from a Google embed – this is online content that has not actually been preserved. But there is a disruptive cookie consent overlay which keeps reappearing after attempting to click on any type of consent.

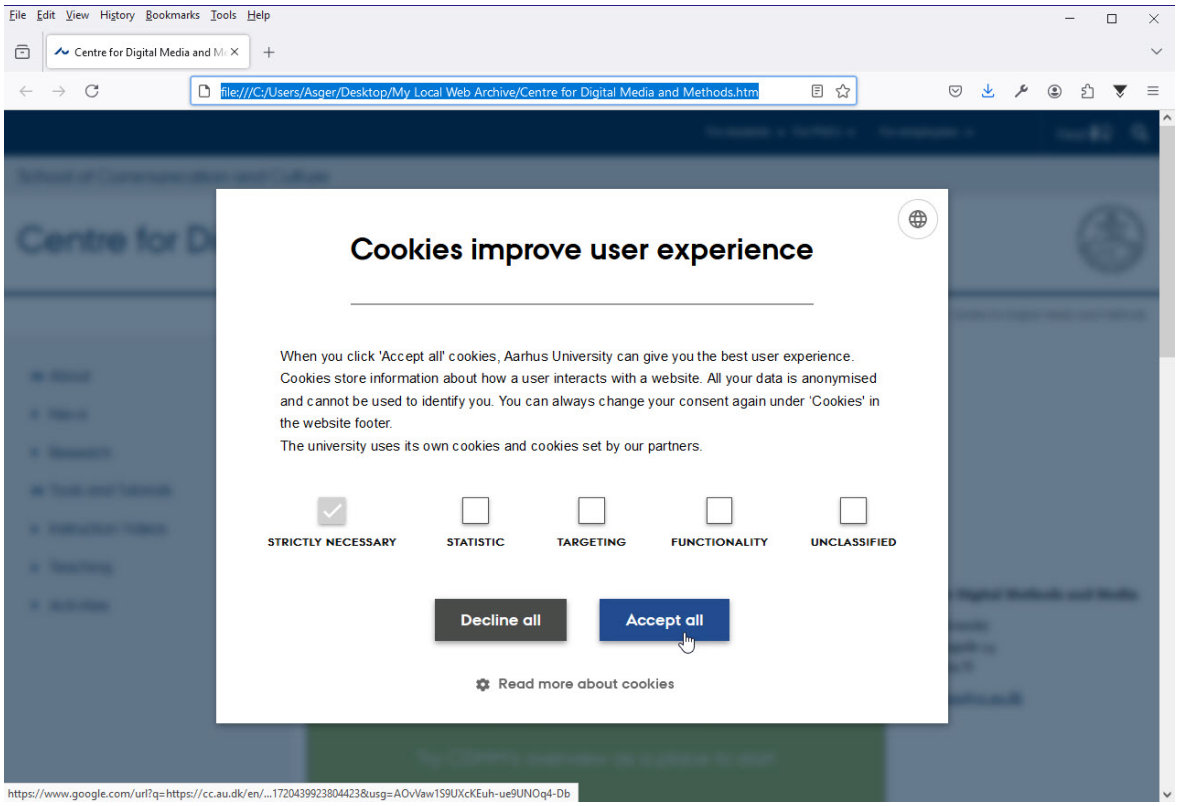


Figure 61: Page saved directly with ctrl+s.

Therefore, if one wants to get a separate folder with the files for the page this method can be used. But in this case where a cookie consent overlay is disruptive for the page itself, a readable copy must be obtained as another version and by other means.

The most stable result from a direct browser save, without embedded content (see 5.1 Challenges for Web Crawlers, p. 79) or cookie consent pop-ups, is usually obtained by selecting “Html only” or similar. The exact names for the saving options may differ from browser to browser. However, if saving more copies to the same folder this can result in one copy overwriting another, wherefore the file name should be changed:

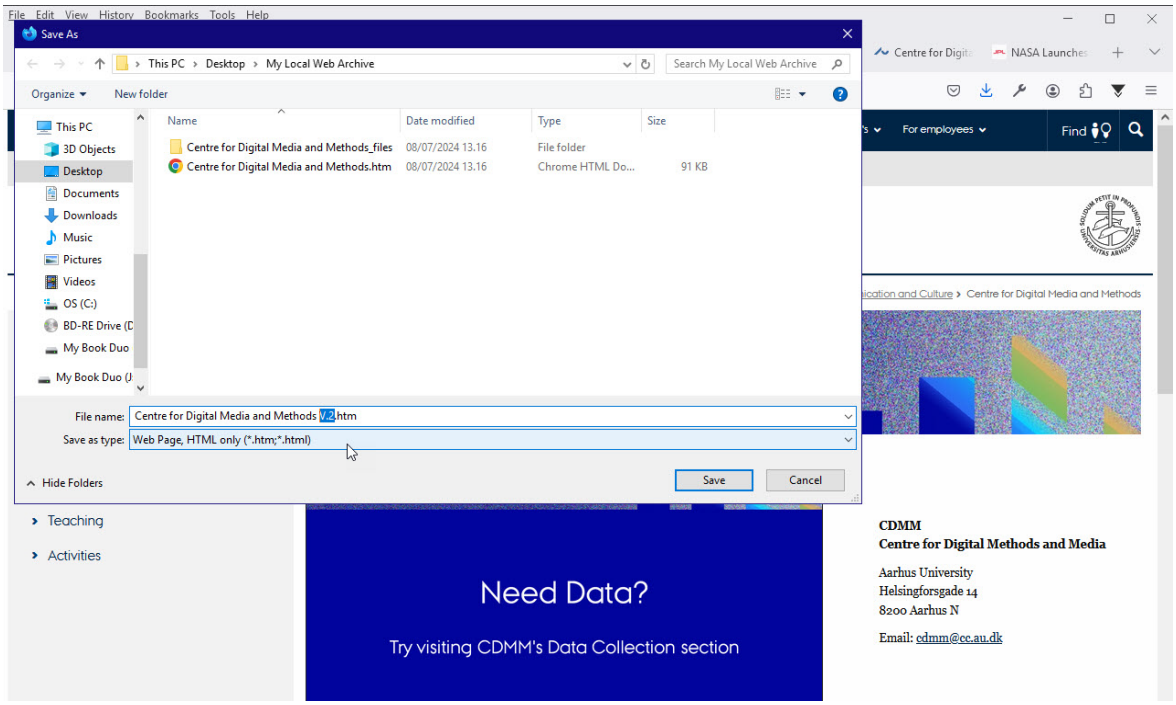


Figure 62: Saving the page as a single HTML file.

In figure 62 a new copy is being saved to the same folder as the previous copy, using the option “Web page, HTML only”. Since the file would otherwise overwrite the first HTM file, the file name is changed with a small addition, in this case “V2”. This is the result of the HTML only save:

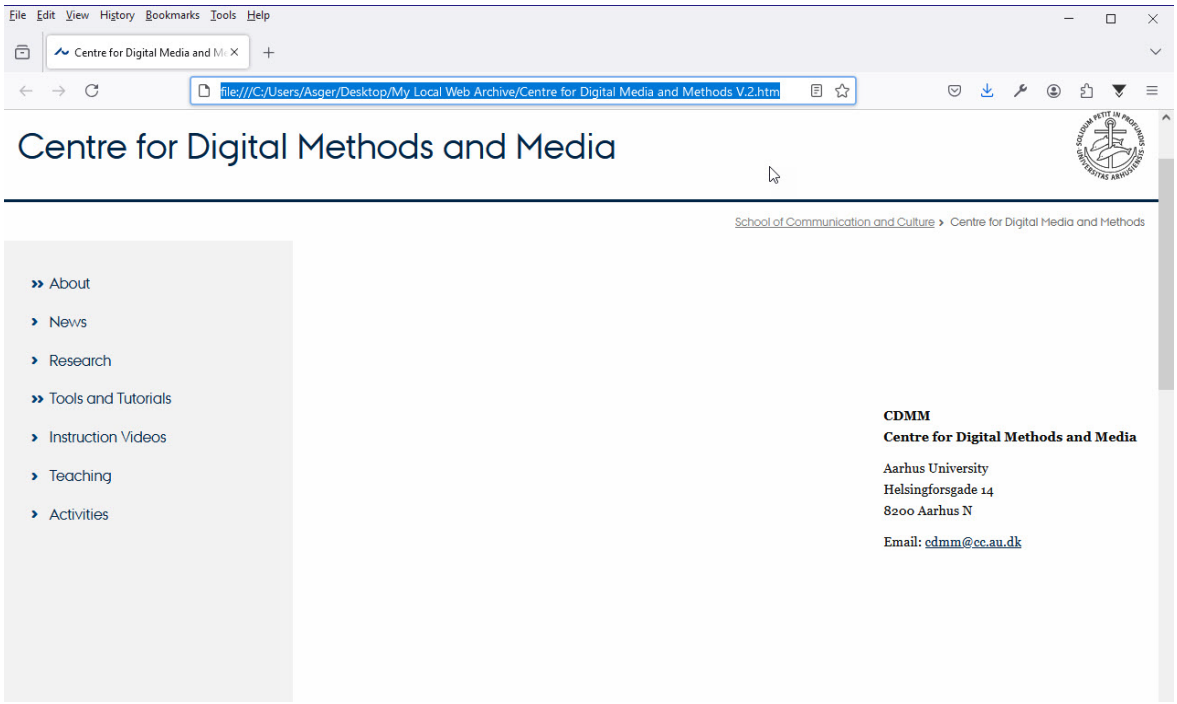


Figure 63: Viewing the HTML only copy.

The page is readable, with text from the website itself preserved. The Aarhus University logo is an embed delivered from an online resource, and this is not disclosed by mouseover. Please see 9.4.3 Checking the Quality of a Harvest, p. 216-19 for explanation and further details, and 7.6.3 Checking Against Online Leaking, Local Archives, p. 112 on how to inspect archived content with online leaking prevented.

The primary disadvantage of this method is that the saved copy does not appear exactly as the original on the live Web, mostly because any sign of an embedded slideshow is missing from the large blank space.

The third technique is to find a way to save as much content as possible in an HTML file by using a third party application. Presently such an application exists, but is may only serve as an example since it may cease to do exist, or stop working due to changes in browser or websites technologies. But there are good chances that similar functions will be available, so it is worth looking at an example.

In this case the example is SingleFile, a powerful browser add-on (in 2024), supported by several of the browsers that are presently widespread and popular, e.g. Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari.

The best way to get it is by opening one of these browsers, go to add-ons/extensions, search for SingleFile, and add it to the browser. If this application no longer is available, the next option would be to search the add-on or extension section for “save complete page”. This should hopefully result in alternatives.

With the SingleFile extension added to a browser, a page can be saved to a single HTML file by right-clicking (ctrl+click on MacOS). One may have to find a blank section without any interactive elements before SingleFile will appear in the right-click context menu.

Several options are offered by this add-on, but here we will focus on getting the best possible local copy of a web page, by right-clicking

and choosing “Save page with SingleFile” from the SingleFile menu entry:

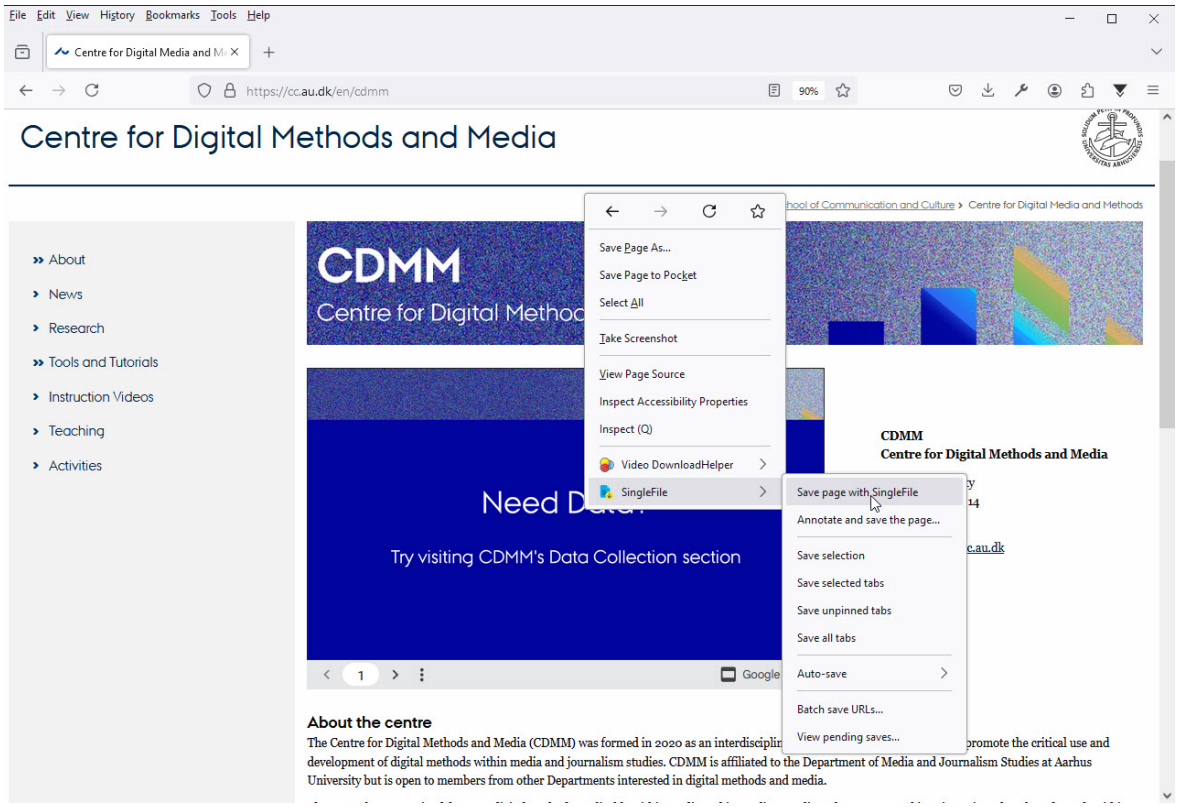


Figure 64: Saving a page with the SingleFile add-on.

By default this add-on does not offer ways to change the file name or the save location. It will be saved to the “Downloads” folder on the computer with a predefined name consisting of the page title and a timestamp, and must be moved manually to any other collection folder. These defaults can be edited by advanced users in the add-on’s “Options” section if needed.

The result from saving the web page with SingleFile is interesting in several ways:

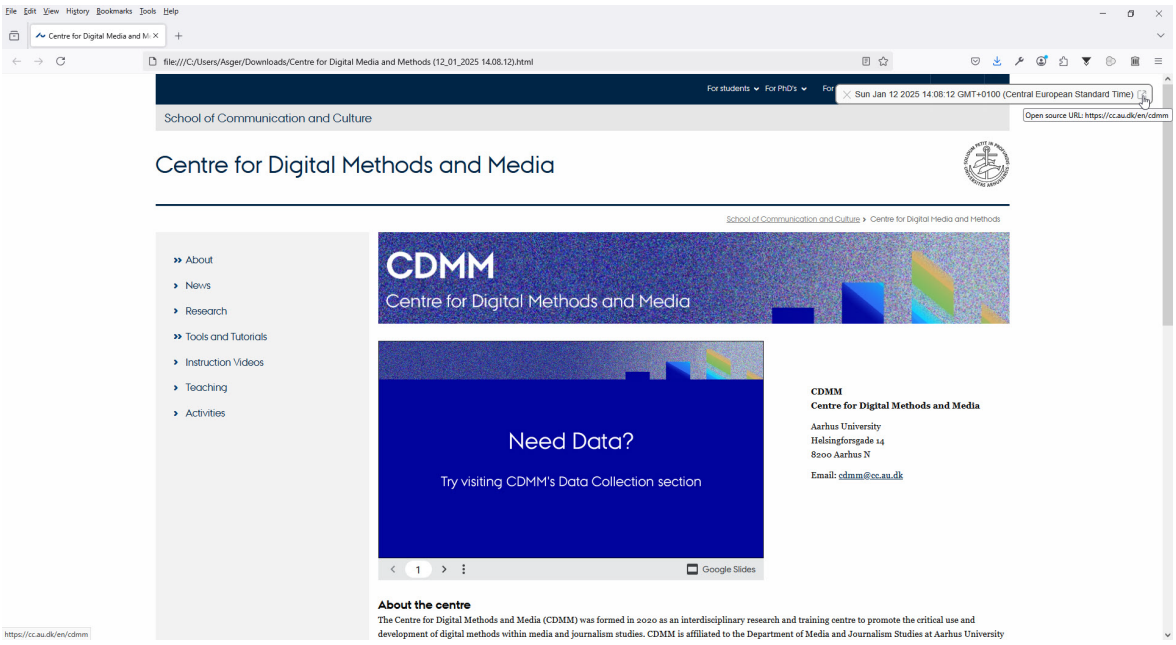


Figure 65: Viewing the version saved with SingleFile.

First and foremost this is the best direct and local save in respect of capturing the layout and content of the web page. If tested in an offline environment as described in 7.6.3 Checking Against Online Leaking, Local Archives, p. 112, the Aarhus University logo, which resisted direct copying in the previous attempts, has been preserved in the archived copy. But as with all archiving of online content, the copy still comes as a version with changes:

The embedded slideshow has been captured in the sense that one can see that it is there, but it does not change automatically, and neither will the forward/backward arrows bring in another slide. Only the slide that was active when taking the copy has been captured. In that sense this copy of the page is still incomplete, because it does not include the full embedded content, but it is closer to the online version in content and layout than any of the previous attempts to save it.

An information icon – a lowercase “i” in a circle – has been added at the top right corner. Clicking it reveals a timestamp for saving this copy, a redirect link the original URL for the web page, and the option to close this additional information (see figure 65).

In summary, there are several ways to save a page directly to a local copy, but all archived copies will be versions with some changes in respect of the online originals. All saved versions will have advantages and disadvantages, and combining more save types may be the best way to ensure having the data which one needs.

9.2.2 Screenshots and Screen Recordings

Screenshots and screen recordings (video captures) are treated together because they have three important similarities:

They capture the exact look of the original content, they do so at the cost of getting machine readable text and working functions such as links, and the methods for creating them are much alike.

Screenshots are useful in several ways. If or when something cannot be saved fully or perfectly, screenshots can serve as additional documentation. They are also useful as illustrations in articles, reports, presentations, etc. But due to the loss of machine readable text, as well as links and functions, they are not ideal as research data.

Screen recording or screen filming is similar, but of course cannot be used directly for printed texts. Their primary value is as documentation of content that cannot be saved otherwise, e.g. live streams on social media, or interaction with advanced web pages or websites with functions that cannot be archived. Screen recordings can serve as proof of how interactions took place, or as demonstrations in presentations.

There are many ways to create screenshots or screen recordings. Some of these are included in Windows or Mac OS, but the systems offer different options which sometimes change, wherefore readers will have to search for the methods relevant for the present version of their present operative system. For example, from Windows 10 and forward an “Xbox Game Bar” may be called, and can usually be used for screen recording although it is dedicated to specifically work for games. This application’s interface and functions have changed several times, and may do so again in the future.

There are also third party applications that can handle screenshots or recordings very well; some are free while others must be purchased. Please refer to 10 Searching for Software and Services for advice on finding relevant software.

One particularly relevant way of taking screenshots of web pages, is in-browser screenshots. A screenshot function is built-in in some browsers, e.g. Mozilla Firefox where it is activated with a right-click (ctrl+click on Mac OS), in other browsers it can be obtained by installing an add-on.

How to take screenshots of web pages may differ from browser to browser (if supported), but in most cases the user is given a choice of capturing the part of the page that is visible, the entire page, or a section that the user selects manually.

If saving single web pages it is worth considering also taking an entire page screenshot, in order to ensure that all content has at least been captured visually, and that a suitable source for an illustration exists.

9.2.3 Saving to Documents

Saving single web pages to documents is usually not an attractive way of archiving web content, but it exists and it can have its uses.

It is an easy way to preserve main content from web pages – text, URL references, and to some extent images – and it can also be used for copying the text content of posts in social media.

The look and feel of the original content will be lost, since the text and images will be forced into the document format and layout.

Saving to documents is thus mostly useful if the text is the most important element, or if a document version is needed for sharing or collaboration, or as an attachment in a full scientific document.

There are two primary ways to save to documents:

1) The content of a web page may be marked and copied, then inserted into a document of any type. Simpler document types such as the notebook format (.txt) do not support active URLs or images, wherefore office type documents are usually preferable.

   Please notice that the source URL for the content must be copied and inserted manually in the document copy or in a log in order to be preserved.

2) Content may also be saved or “printed” directly into PDF files. Depending on one’s browser and operative system this can be done in a number of ways:

On a web page press the shortcut for printing (ctrl+p in Windows, cmp+p in MacOS), and a number of options should become available, including physical printing on paper or printing to a PDF format.

Operative systems may have a built-in PDF printer, but usually this option will cause active links to be lost, and also may not include options for setting the “paper” format. A problem with content being “cut” and incomplete at page shifts is also likely to occur. Disturbances from page shifts are a general problem with PDF conversions, as described in the following example:

Presently (2024) browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge have a built-in PDF converter, and more browsers may also have such a feature. The conversion quality may vary from browser to browser, and from web page to web page. Here Google Chrome is used as an example:

In Google Chrome pressing the keyboard shortcut for “print” opens an overlay window with printing options. Selecting the browser’s built-in PDF converter (called “Save to PDF” in all browsers mentioned), rather than physical print or other PDF converters on the computer will ensure that, a) links will be preserved as active hyperlinks in the PDF

copy, and b) by marking the relevant option the source URL for the web page will be inserted in the document.

At the bottom of the settings, presently under “More settings”, one can select that the page source should be included by marking “Headers and footers”.

One may also choose between paper formats. The A4 format will probably be preferable if the copied web page is intended to serve as an attachment in a larger document, but it is also possible to select other standard sizes, vertical or horisontal paper direction, number of pages per “sheet”, or custom scale.

One can experiment with the settings, and see a preview of how the resulting PDF document will look. Unfortunately the aforementioned problem with images or text being cut by page shifts, with a resulting loss of one or more lines, is hard to bypass in PDF conversions.

Here is an example of an attempt to convert a web page to PDF with Google Chrome (2024):

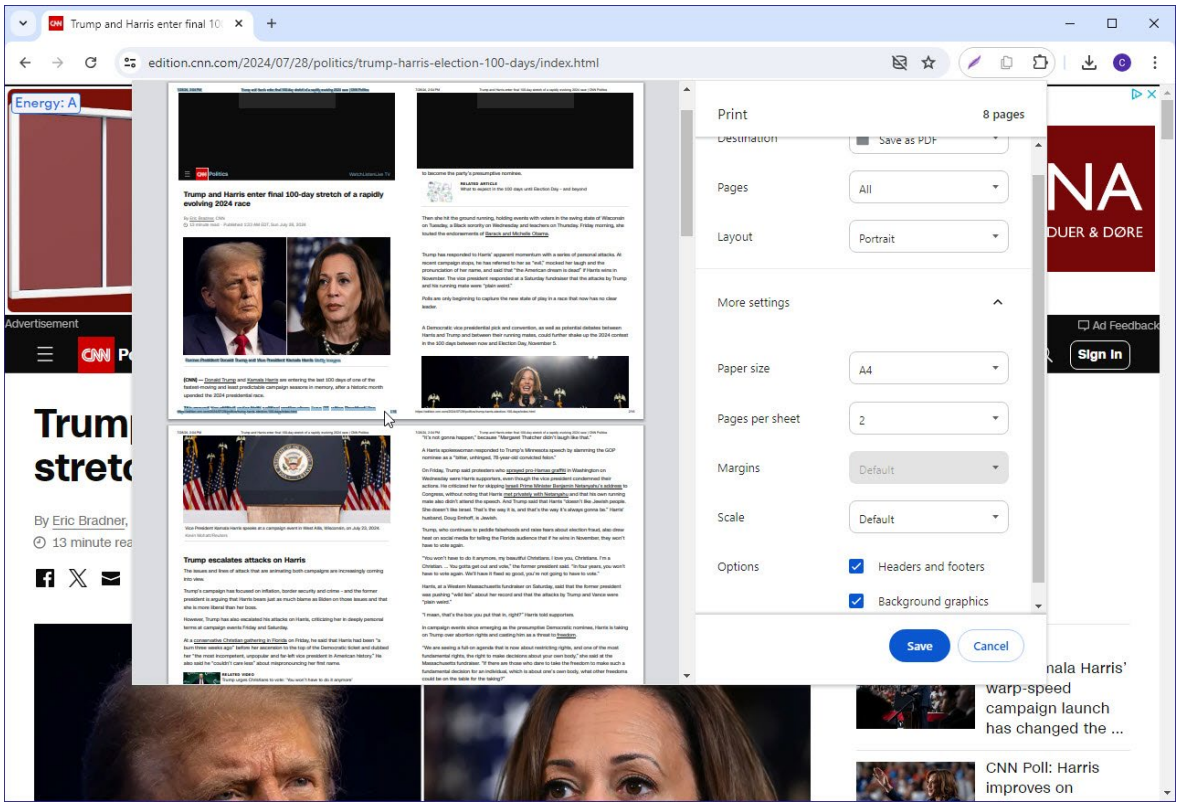


Figure 66: preview of a PDF conversion with settings options.

Note that “More settings” has been opened, and that “Headers and footers” has been marked. In the preview a date stamp and the title of the web page has therefore been inserted at the top, and the URL for the web page at the bottom.

Page shift detail view:

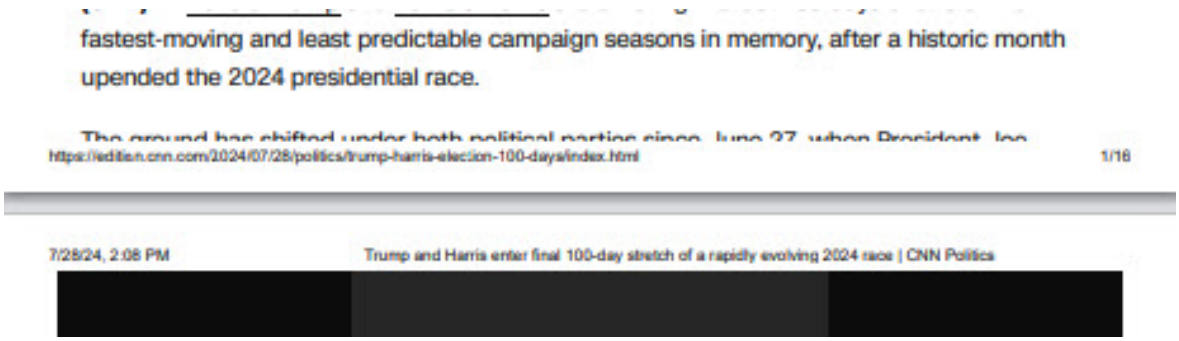


Figure 67: PDF conversion has a tendency to disrupt text and other content at page breaks.

Just above the URL at the bottom of the first page is an example of a paragraph that has been cut and partially lost due to a page shift.

The purpose of this example is to illustrate the potentials as well as the limitations of most forms of PDF conversions.

If document copies should be of interest the reader is advised to look for add-ons for making web pages “printer friendly”. These types of add-ons are normally available, and with luck the reader may find one that will allow custom adjustments of the content so that it will fit into a document without being cut. See also 10.6 Browser Extensions.

9.2.4 Archiving at The Internet Archive (or Alternatives)

With a few limitations it is possible to archive single pages directly at The Internet Archive, or the Portuguese Web Archive; Arquivo.pt.

Such copies are highly useful as documentation or as references. They are of course neither local nor controlled by oneself, except that they are created directly upon one’s request. Saving to a large open archive has the advantages that, a) this generates archival pages for reference from known and accepted archives, and b) the pages will

coexist with the entire archive of websites and web pages that may serve contextually for the added copy.

It is realistic that one can find services that can create copies of scripted pages that resist standard harvesting techniques (see general advice in 10 Searching for Software and Services. This approach may be considered for pages that cannot be saved directly to an institutional archive – but if this is done from a personal account that permits sharing, one must consider the legal and ethical framework for how and if the content may be shared for use as a reference, see 11 Legal and Ethical Concerns.

Copies of pages from different resources will also call for attention to standards of referencing, please see 8.3 Referencing from Web Archives.

For details on the “Save Page Now” service at The Internet Archive, please refer to 8.1.2 The ‘Save Page Now’ Service.

9.3 Copying URLs

When looking for web pages or websites in institutional archives, or archiving entire websites oneself, getting and using the correct URLs for web pages and websites of interest can be important.

Normally and for most users copying a URL is a rather straightforward process, but there are a few details that can sometimes become obstacles.

As mentioned in chapter 4.1.2 URLs: Uniform Resource Locators, a correct URL should sometimes include the prefix “www.”, and sometimes not. Also the protocol call should either be HTTP:// or HTTPS:// depending on whether a website used a secure connection on the live Web.

Furthermore, if a URL is copied or accessed from an indirect resource such as, newsletters or posts found on social software, it is likely to contain tracking information.

The reason why this calls for attention in connection with web archives or web archiving is a question of need for precision, versus normal usage where copying and using URLs does not require anything but copying and inserting and saving, or sending.

In a URL list obtained from an authoritative resource, such as an administrative institution for a top level domain, one may already have correct URLs. But when searching for resources on the live web, or attempting to retrieve or make archived copies, it is sometimes necessary to obtain the relevant URLs in a precise and specific way.

Developers of browsers, as well as developers at institutional web archives – are aware of the problem that users may not always get the protocol right, and may omit the “www.” prefix when it should be included, or vice versa. Therefore most software and services will be able to find the correct results if the domain address itself is correct – but tracking information may still cause errors.

But there are exceptions from the rule that a web address without all the URL prefixes will normally work. For example, the web harvester HTTrack is a free and longstanding software for use on personal computers which is still available (in 2024) – and it will only harvest web pages correctly if given a precise URL, with the correct protocol call and inclusion/exclusion of “www.”.[30]

The problems that readers may occasionally encounter can occur in the process of visiting the Web and copying URLs, for documentation or archiving purposes.

[30] For more on HTTrack the reader may check if it is still listed in Data Collection in the URL for CDMM’s Tools and Tutorials given at the start of 10 Searching for Software and Services, in which case more information is provided there.

As an example we can look at the websites for two major Danish broadcasters, DR and TV2, and how two different browsers handle their web addresses in 2024; respectively Mozilla Firefox, and Google Chrome. This will serve as an example of differences that may occur between browsers in general.

A modern user may enter the domain and top domain for either broadcaster in a browser, like this:

dr.dk

tv2.dk

What happens in Firefox (2024) is that the address is visibly filled out in the address line when going to the requested website. The address for dr.dk visibly changes to https://www.dr.dk/, which is the precise URL for the main page on that website. In Chrome (2024) when entering “dr.dk” and going to the website the address line continues to show “dr.dk”.

When entering “tv2.dk” in the two browsers the same thing happens. In Firefox, the address visibly changes, this time to https://tv2.dk. So in both URLs the correct protocol call is HTTPS, while the “www.” prefix is a part of the correct URL for dr.dk, and this is not the case for tv2.dk.

The same thing happens in reverse, so to speak, if one enters the addresses with the “www.” prefix:

Firefox visibly changes www.dr.dk to https://www.dr.dk, and www.tv2.dk is visibly changed to https://tv2.dk. In Chrome, “www.” is similarly kept or removed and the correct protocol call is added, but this is not visible; only dr.dk or tv2.dk will be shown.

Whether the browser shows the exact URL and whether this is preferable is mostly a matter of taste. The important thing is to be aware that the safest way to get a visited URL completely right is by copying it from the browser’s address line.

If the web address is copied from the address line, the copied text will show the full URL when inserted elsewhere. In this example, the full URL for dr.dk is obtained by copying the address line in Chrome and pasting into a text file:

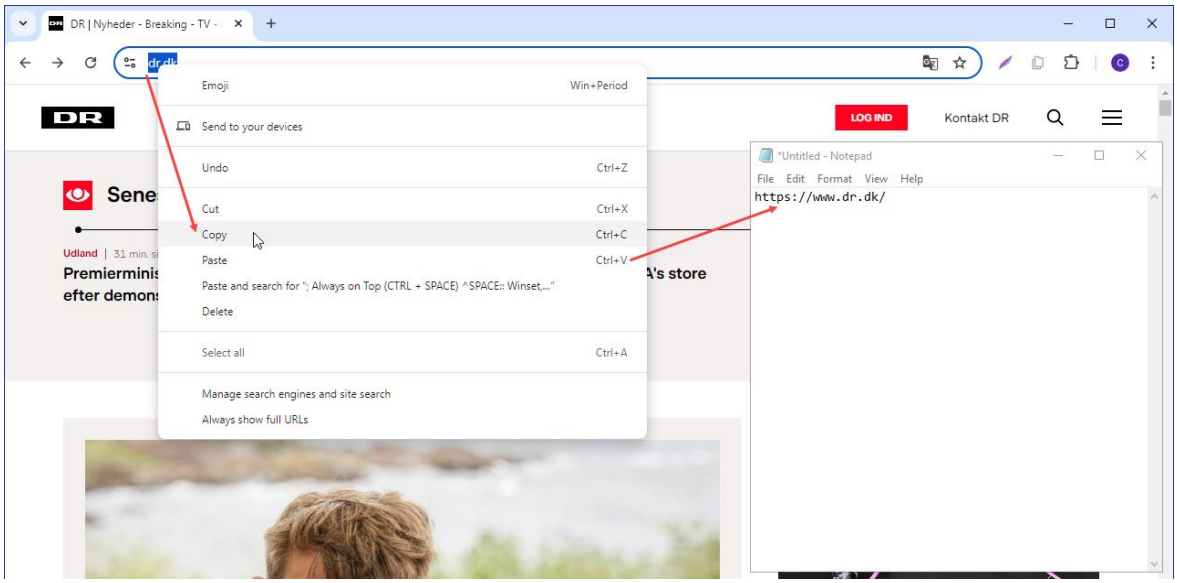


Figure 68: Copying from a browser's address line obtains the full URL.

9.3.1 Tracking Information in URLs

In some cases a user may come to a web page from an external source, such as a linked reference found on another website, in a post on a social medium, in a news email, or from a shared link of said type from a contact.

More often than not, a URL from such sources will contain tracking information, so the website owners can monitor how, and from where, their visitors find their content. This interest is legitimate, and may or may not contain information that will identify a specific user. For example, a URL with tracking information showing that "the user came here from a link found in an open article" will identify the source of the article, but not the user. On the other hand, a link copied when logged in to a specific user account, or from a subscribed newsletter may contain information on the account or subscription the URL came from.

The questions of how and to which extent tracking is used for identifying persons, and thus for example target advertisements based

on patterns of personal interests – are interesting, but this is not the focus here.

When working with sources and their URLs for research, archiving, or archival purposes, the tracking information that may be found in URLs is important to be aware of as a form of noise that should be taken into account.

As an example, here is a URL from a NASA article on a new climate observation satellite as it was linked in a newsletter from NASA's Jet Propulsion Laboratory on June 5, 2024:
https://www.jpl.nasa.gov/news/nasa-launches-second-small-climate-satellite-to-study-earths-poles?utm_source=iContact&utm_medium=email&utm_campaign=nasajpl&utm_content=prefire20240605

All the information beginning with a question mark in this URL is tracking information. This lets the website owners know how the reader found the article, e.g. from a newspaper article, or in this case by somebody actively making use of their newsletter.

From an archiving perspective, the problem is that the tracking information does not "belong" directly to the page specified in the URL. It is added information and this added information may cause confusion; a) in web harvester programs, which may or may not be able to handle a URL with tracking information, or b) when looking for web pages in existing archives.

Using the URL address in the example above for a search at The Internet Archive *with* the tracking information yields this result:

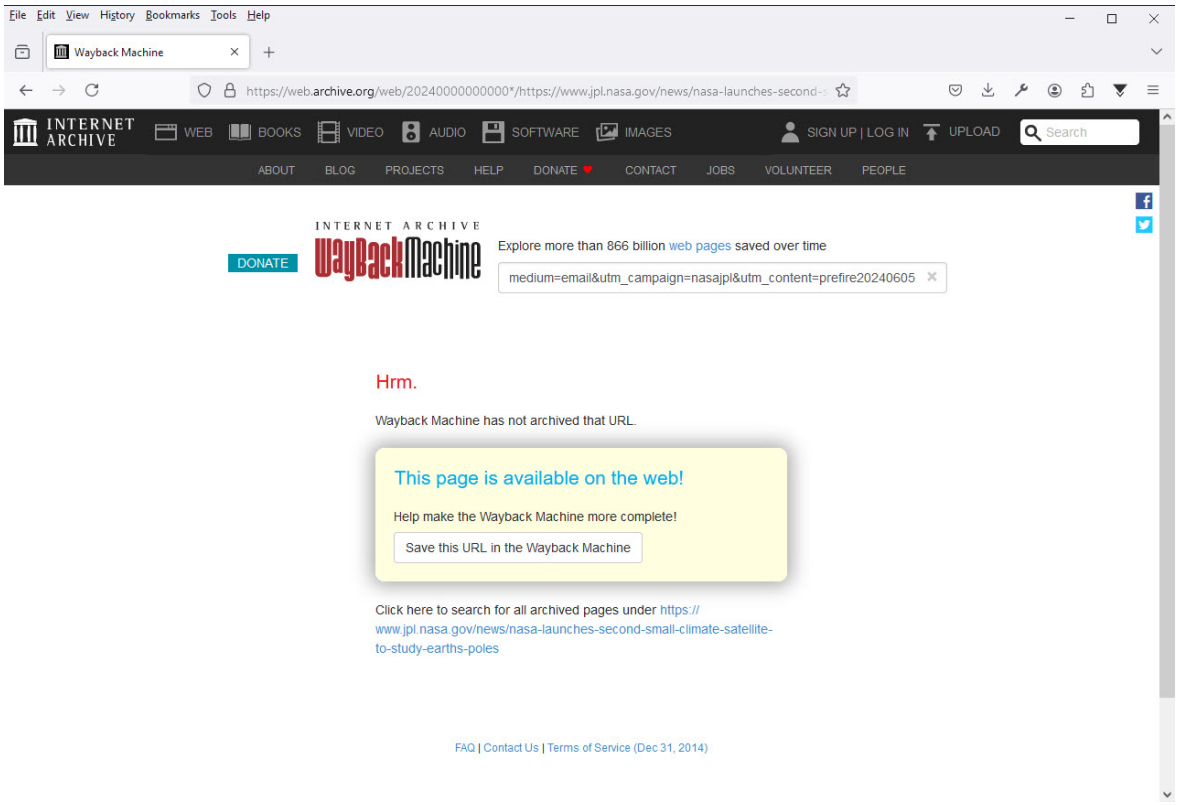


Figure 69: Using a URL with tracking information yields no results.

This may cause the archive's visitor to assume that the relevant page has not been archived. But this is not correct. If one shaves off the tracking information starting with and including the question mark, one gets a cleaned URL which points directly to the web page in question.

In the NASA example, the result is as follows: https://www.jpl.nasa.gov/news/nasa-launches-second-small-climate-satellite-to-study-earths-poles

Entering this corrected and direct URL in a search on The Internet Archive yields an entirely different result:

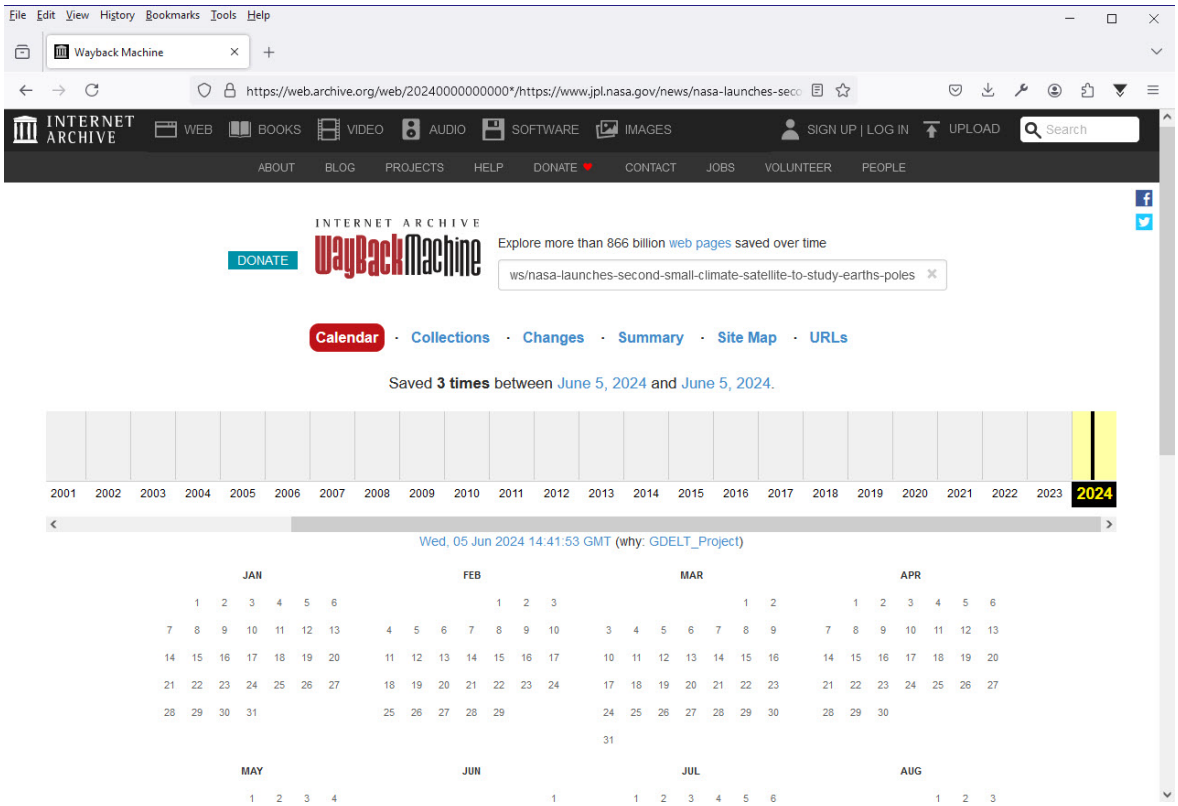


Figure 70: To find an article in an archive, the URL must be clean of tracking information.

All of the three captures found in this search result (on June 5 2024, not shown in the picture) lead to good copies of the original page.

So for archiving purposes when attempting to harvest websites oneself, and for archival purposes when attempting to find copies of resources in existing archives, one should pay attention to removing the tracking for a given URL.

"Given" is the operative word here, because the problem only occurs if the URL was given from an external source. When visiting a website directly and moving around on it via the menu or other internal links, the URLs will be direct. Tracking information signifies an external resource, and its kind and origin.

To complicate matters slightly, a question mark may sometimes be a part of the URL syntax signifying a query for a specific object. In such cases the questions mark and the information following it cannot be removed without the adverse effect of not getting the actual target.

As an example of this, here is a YouTube URL for a video on mapping an exoplanet's weather using the James Webb Telescope, linked in SETI Institute's Weekly Newsletter, June 6, 2024: https://www.youtube.com/watch?v=rNMWzLItJ4I&list=PLw6IJozmaWbTt2pRHTFS6ySZmSr97-Wcd&index=2

The information after the question mark in this example give the video ID and the playlist that the video is a part of. In this case, removing everything beginning with a question mark would result in a visit to YouTube's front page, and not the desired video and its page.

This means that consistently removing everything after a question mark in a URL, e.g. by doing this automatically from a long list of URLs – is not a good idea. But usually it is easy to spot where a URL with domain and subpages ends and the tracking information begins, always signified with a question mark.

At the time of writing this, most browsers do not offer a feature allowing users to copy URLs directly without tracking information if this is present. But Firefox and Firefox Developer from the Mozilla Corporation do in 2024, as illustrated here:

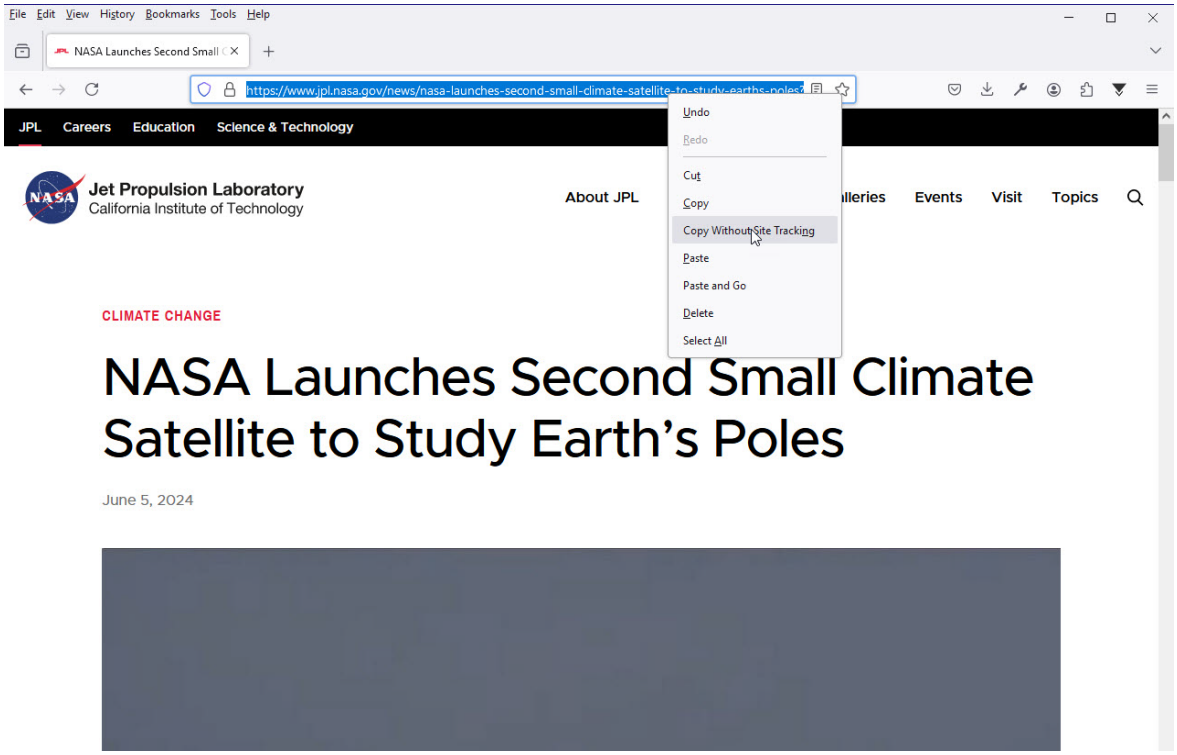


Figure 71: Mozilla Firefox (in 2024) allows for URL copying without tracking information.

It is very likely that this feature will spread to other browsers over time so this information is not static; it is only to make the reader aware that the option exists and may be useful.

Readers who use browser extensions for copying URLs (as mentioned in 10.6 Browser Extensions) must expect that any URL opened with tracking information will normally also be copied with it, unless the URL copy add-on is used in combination with an add-on that removes tracking information automatically.

9.4 Archiving Websites or Specific Content from Websites

If or when full or systematic harvesting of one or more websites becomes relevant, then one will need an application that is capable of harvesting and saving websites to offline copies. It is recommended to read chapters 4 The Web as a Technology and 5 The Web Archiving Process before starting to archive websites, in order to better appreciate what a web harvester does and how one can work with it.

9.4.1 Web Harvesters vs Web Scrapers

Two types of web crawlers are relevant for website harvesting; web harvesters or web scrapers.

Web harvesters are applications designed and dedicated to preserve web content by saving files from web pages and websites, in a manner that allows for a “born again digital” replay.

Web scrapers can also do this, but they are also meant for analytical data extraction. Web scrapers are the common name for paid software and services that are primarily meant for businesses and organisations with an interest in search engine optimisation. In respect of search engine optimisation it is relevant to businesses and organisations to know, e.g. where it is good to be mentioned or how pricing develops on their relevant market. The abbreviation for search engine optimisation, “SEO”, is often included in the brand name of these services. Web scrapers can be set to extract specific data from websites, such as pricing or outgoing links. Such specific data

extraction cannot be done as easily with a traditional web harvester. And such specific data extraction can be relevant for researchers, e.g. for analysing website networks.

There are many services and applications to be found on the Web, of different pricing and quality. A free web harvester may be perfect for one's needs, if one just wants to preserve a few websites on occasion, and if it is not important to make copies as fast as possible due to rapidly changing content. As commercial products, web scrapers will often be faster than free harvesters.

Both types of programs are web crawlers, and can be set to save entire websites, or to extract specific file types, but refined data extraction from the HTML code is only supported in web scrapers.

The HTML files for the pages in a locally archived website can be opened in a web browser, and the harvested copy of the website can be explored from its menu in the same manner as the online original.

9.4.2 The Folder Structure in Local Archives

When websites are saved to offline copies on a local archive, the web crawler will save the content in a system of folders in a main target folder, which can be changed in the software if one is not satisfied with the default setting. The folder structure of websites is discussed in further detail in 4.1.2 URLs: Uniform Resource Locators, see for example figure 7 on page 54.

The folder system follows the structure of the website so that the main or "home" page for the domain address will be the main folder, the first layer of subpages which are normally listed as main sections in the website menu will be stored in corresponding subfolders, subpages from the main sections will be placed in their subfolders, etc. URLs are changed to reflect the location of the saved files in the folder system.

In this example, the stored content had been placed in a standard folder called "My Web Sites", and the harvest job has been given its own folder called "CFI" (for "Centre for Internet Studies").

In the CFI folder, the main folder reflects the main domain address, cfi.au.dk. In that folder there are subfolders reflecting the subsections, e.g. “Publications”. In that folder there are three files that represent the three pages found in this section; books, memberpublications, and monographseries. The fourth file, “archiving.html” and its corresponding subfolder are for a page that is referred and linked to on the “books” page but not in the menu.

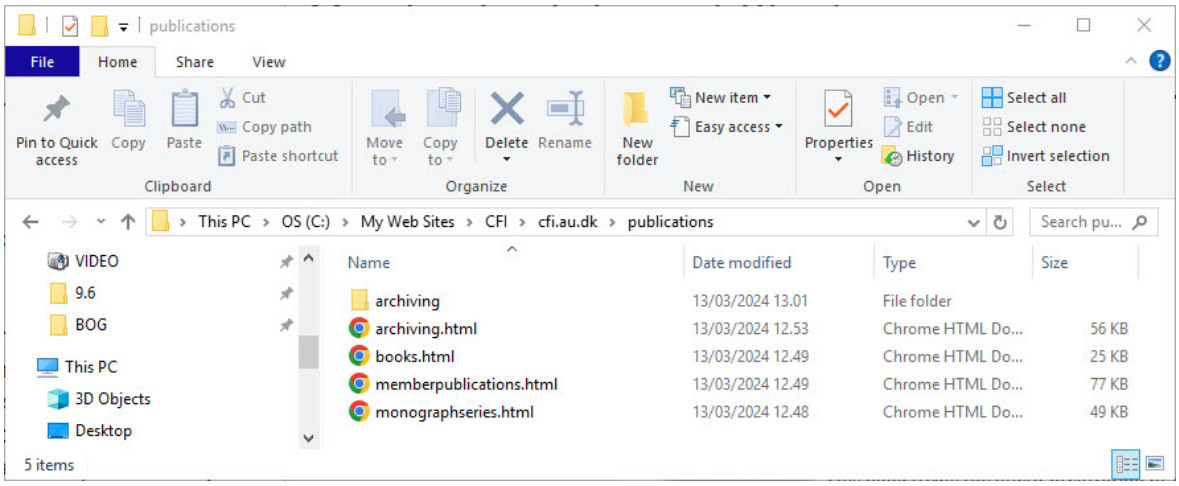


Figure 72: Inspecting an archived folder for the subpage "publications".

As described in the chapter 4.1.2 URLs: Uniform Resource Locators there is a close likeness between the structure of URLs and the folder systems found on computers.

If one marks the address line for the folder above, it very much resembles the address lines used in browsers.

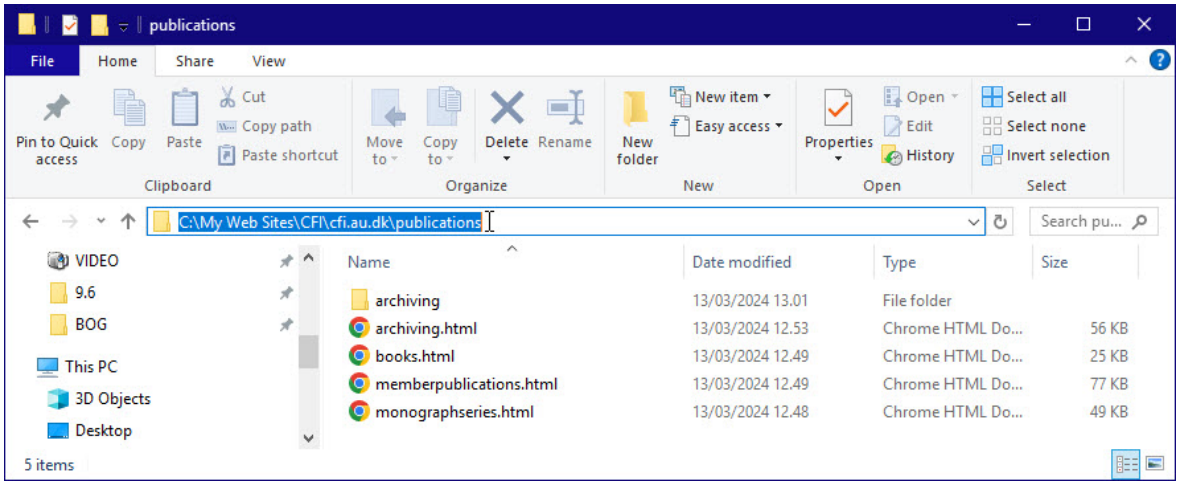


Figure 73: The folder path for the "publications" subfolder.

As mentioned in 5.2 URLs Are Changed, web harvesting results in changes to the addresses in an archived copy of a website. This will preserve the connection between the archived pages, permitting offline replay whereby one may also check the quality and completeness of the harvested copy.

It is worth mentioning that the folder for a harvested website will usually contain additional folders that do not belong to, or reflect, the immediately visible website architecture, but represent various side- or subdomains from which content such as images or files for download are placed onto the website's pages.

By-harvests of other domains will be found beside the main domain folder. The folder for the main domain itself will contain the data that technically belongs to the targeted website's architecture. Both are results of the settings for width and depth in the harvest job as discussed in 5.1 Challenges for Web Crawlers, p. 82-83, and provide files and content for the archived copy of the targeted website that would otherwise be missing.

9.4.3 Checking the Quality of a Harvest

As with all archiving it is important to check the quality after attempting to harvest a website:

- Did the harvest preserve all important pages,
- were images and other files included,
- is there embedded content from the live Web pages that one would like to have, but which has not been included?

Local harvesting can be expected to have a level of completeness much as if all the single pages had been archived manually, but the process has been automated sparing one from much manual work, and the website structure has been preserved whereby an offline copy of the entire website can be explored.

Please notice: Web harvesters and web scrapers come with default settings designed to harvest websites as requested by the user, with the necessary width and depth to normally get the entire website, and only the most necessary amount of by-harvest (see also 5 The Web Archiving Process, p. 73-77, and 5.1 Challenges for Web Crawlers, p. 82-83). The examples from a website harvest shown here in this subchapter were created using a web harvester (HTTrack) with the default settings, resulting in a very good copy that should be satisfactory for most purposes. Users may not have to change any settings for their website harvesting, unless it is necessary for their specific needs to get a higher level of completion. The same harvest job was also tried with a web scraper (A1 Website Download), also using the default settings, and with the same results.

The focus will be on a broader inspection of what other content elements have or have not been archived. An example of an embed causing missing content in the archived version as described in 7.4 Missing Content Elements will also be given.

Quality verification can be done by inspecting a saved website copy and paying attention to whether the links in the saved version point to other content in the local archive version, or to online content. Obviously this kind of manual verification will not be feasible for large scale harvests, where a harvest strategy for ensuring a satisfactory level of content preservation is preferable.

But for smaller scale archiving it is wise to verify the quality, and if larger scale archiving is planned, then a manual sample verification of a couple of copied websites may help determine the settings that should be used for getting a satisfactory overall harvest.

One may start at the main page, and click around on the archived website. The following examples of inspection take place with pages in the “Publications” subsection of the archived version of the CFI website, because they include examples of images.

As long as the content is online and unchanged, the original and live version is of course the best possible reference. So starting there:

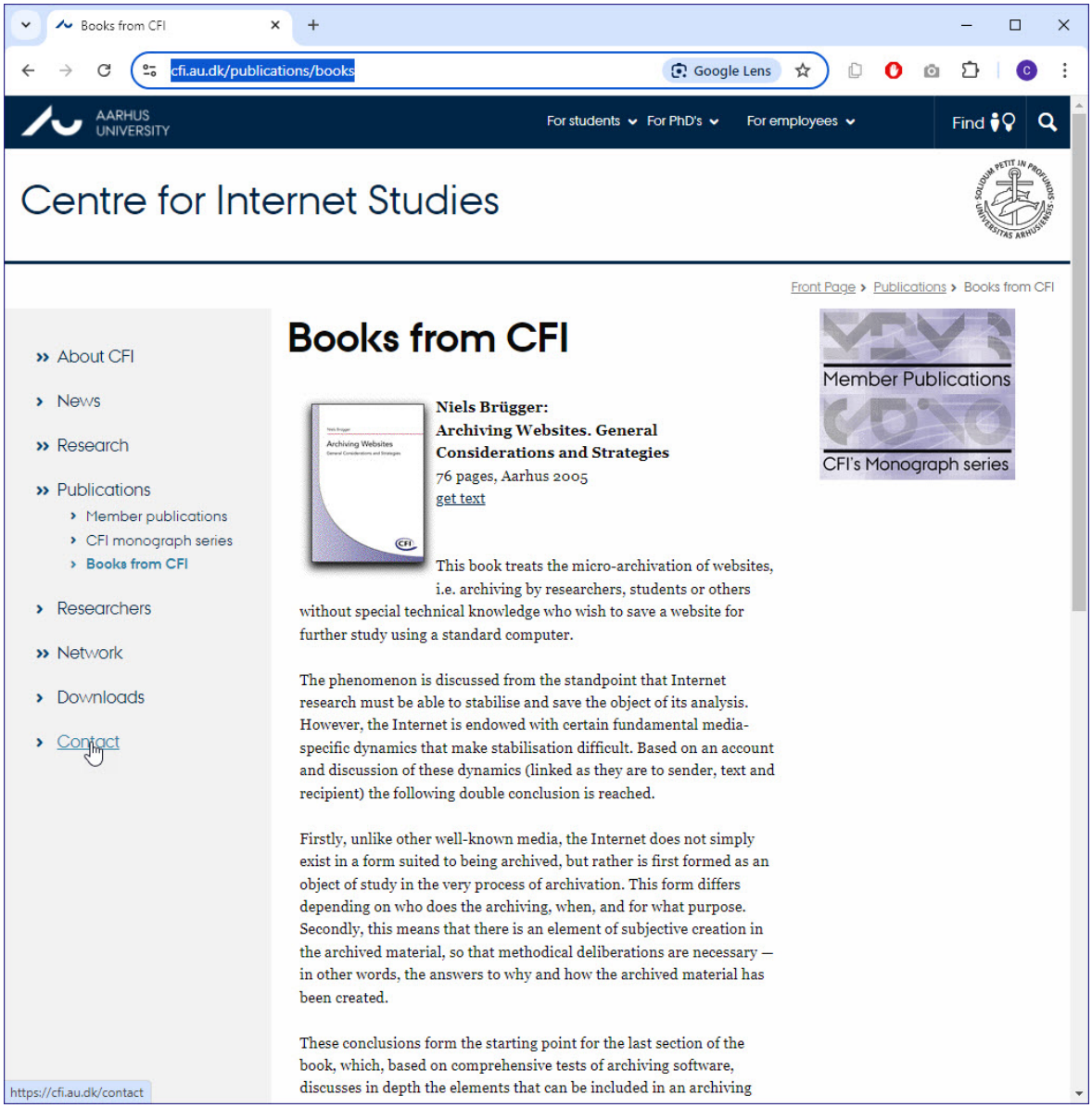


Figure 74: Books from CFI page online, 2024.

The online page has a URL address in the browser's address line at the top. At the bottom another URL is shown as a preview of the address one will go to if clicking "Contact". This is shown by mouseover on that menu point.

In the following example, opening the archived copy of the same page beside the live version confirms that the archived copy of this page seems rather complete.

In the example the hand cursor is once again placed with mouseover on “Contacts”, and an address is previewed at the bottom of the window. This is an offline address pointing to a location in the local folder system, as shown in the examples in chapter 9.4.2 The Folder Structure in Local Archives.

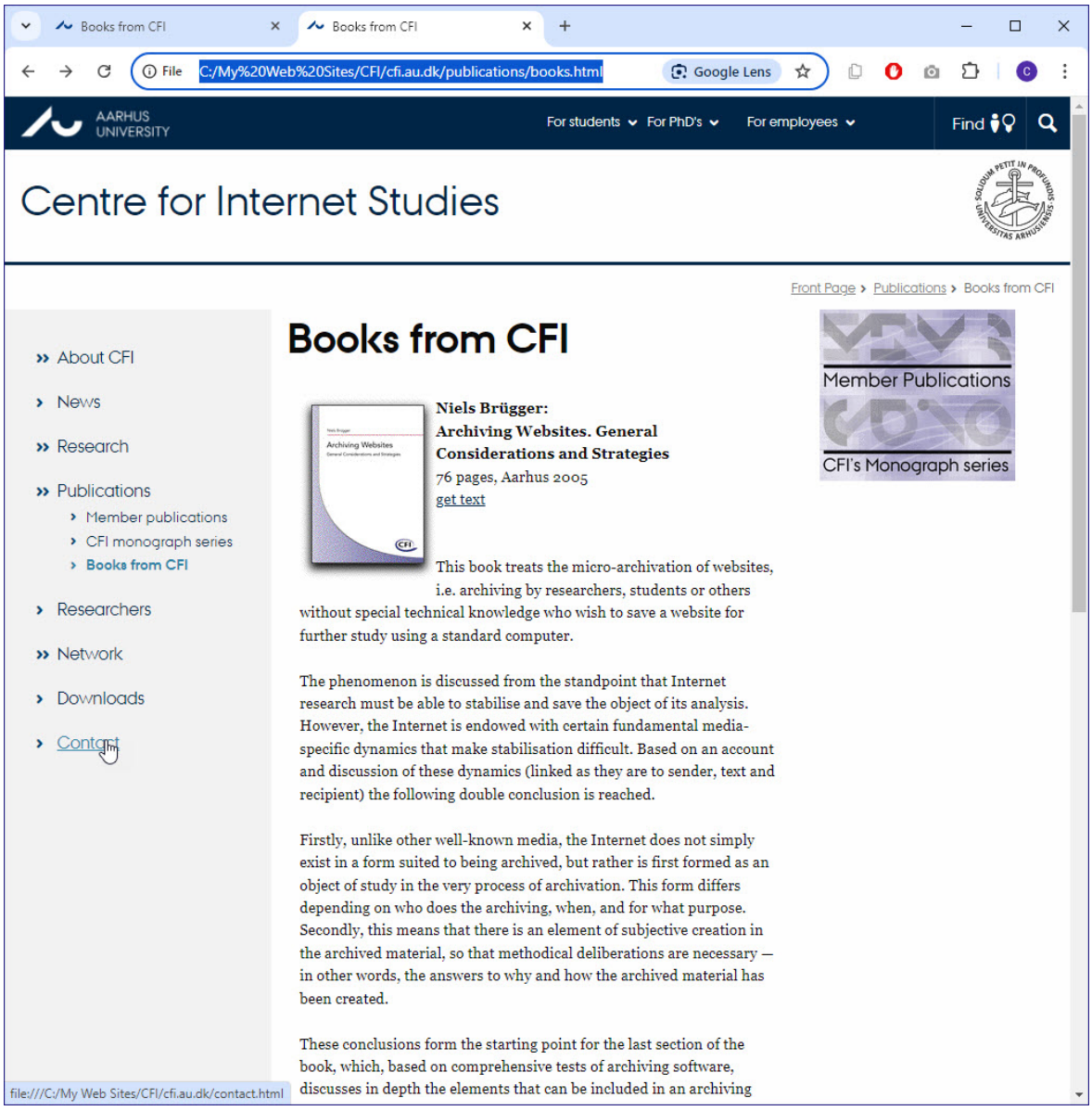


Figure 75: Internal address shown at the bottom at mouseover on “Contact”.

The preview of the folder system address confirms that the page “Contact” is also located in the archive on the computer. The reader may notice that the address line shown in the browser contains “%20” instead of the spaces shown in the folder system in figure 73. This is because some special characters are not allowed in a URL and are

instead represented with character codes, for spaces this is %20. Some browsers will replace incompatible characters visibly, others will hide such changes, but if copied the URL will contain the special character codes instead of spaces, etc.

On the page “Books from CFI” there are various pictures. In the following examples a closer look is provided for a few of them:

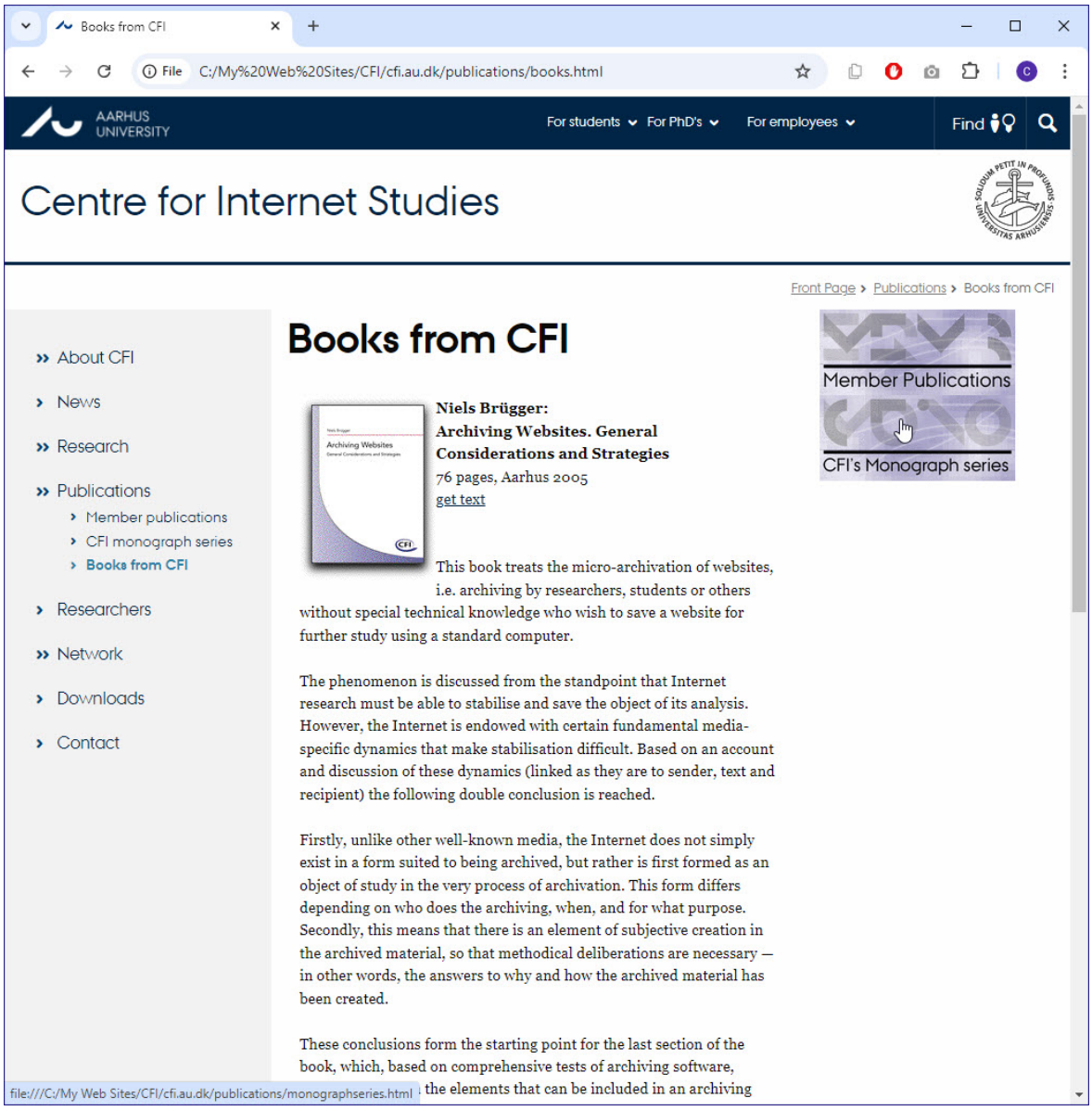


Figure 76: This image points to an archived page.

With mouseover on the lower one of the two images at the right side, “Member Publications”, an address for a local HTML file is shown at the bottom left corner. This yields no information about the image,

besides that it has been set to link to another page (the “CFI monograph series”), and that the archived version of that subpage is located in the folder system. The latter is disclosed by the local address preview at the bottom of the page.

However, it is the picture that is now of interest. Is it saved, or not? Since no information on the picture is provided by mouseover, the next step is to see if there is a link for the picture itself. In the next example the picture has been right-clicked. Notice that there are two links that can be copied: “Copy link address” will copy the address that the image is set to point to, while “Copy image address” will copy the address for the image itself.

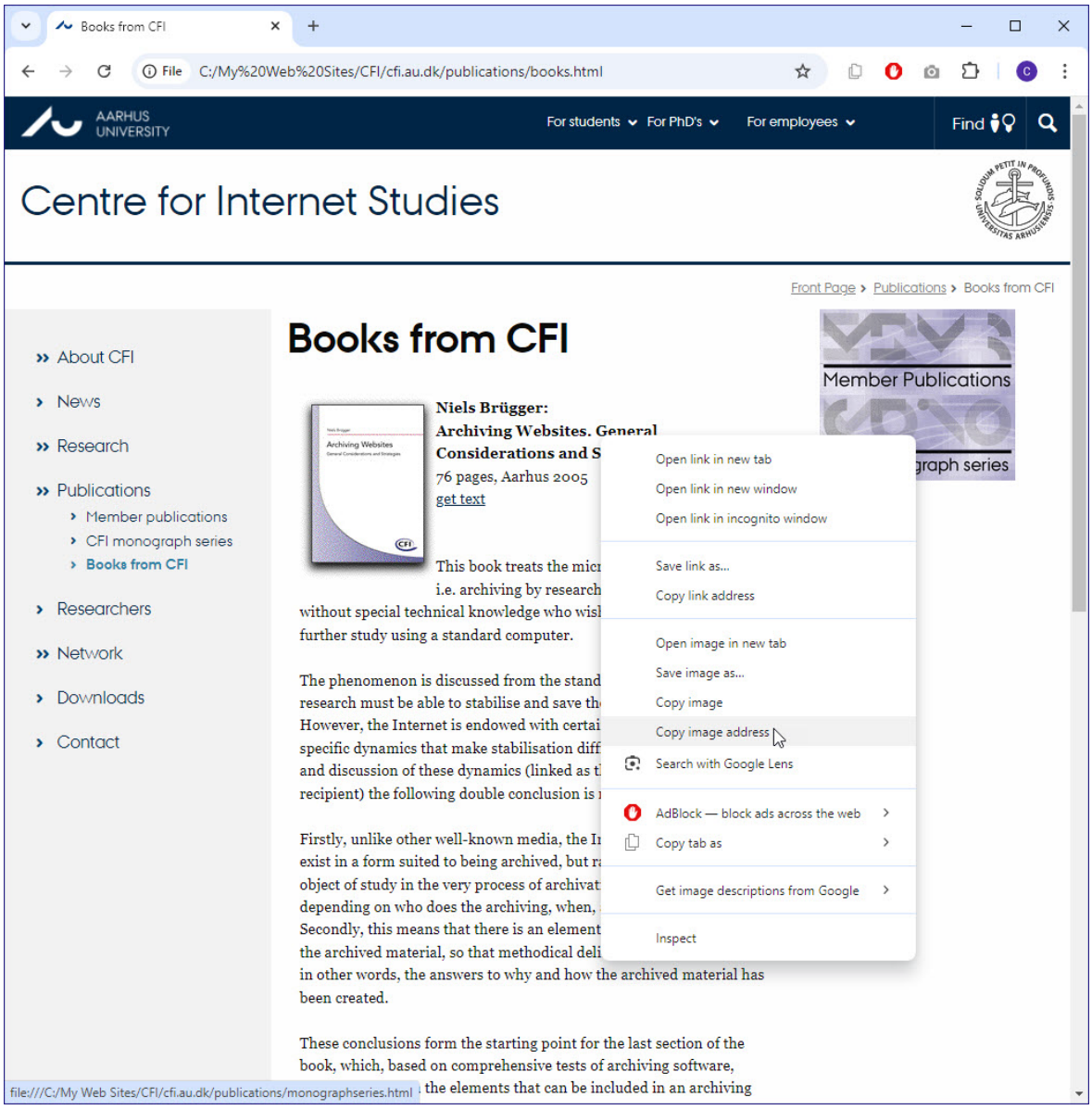


Figure 77: Getting the exact address for an image.

With the image link copied a new tab is opened, and the address is inserted, with the following result:

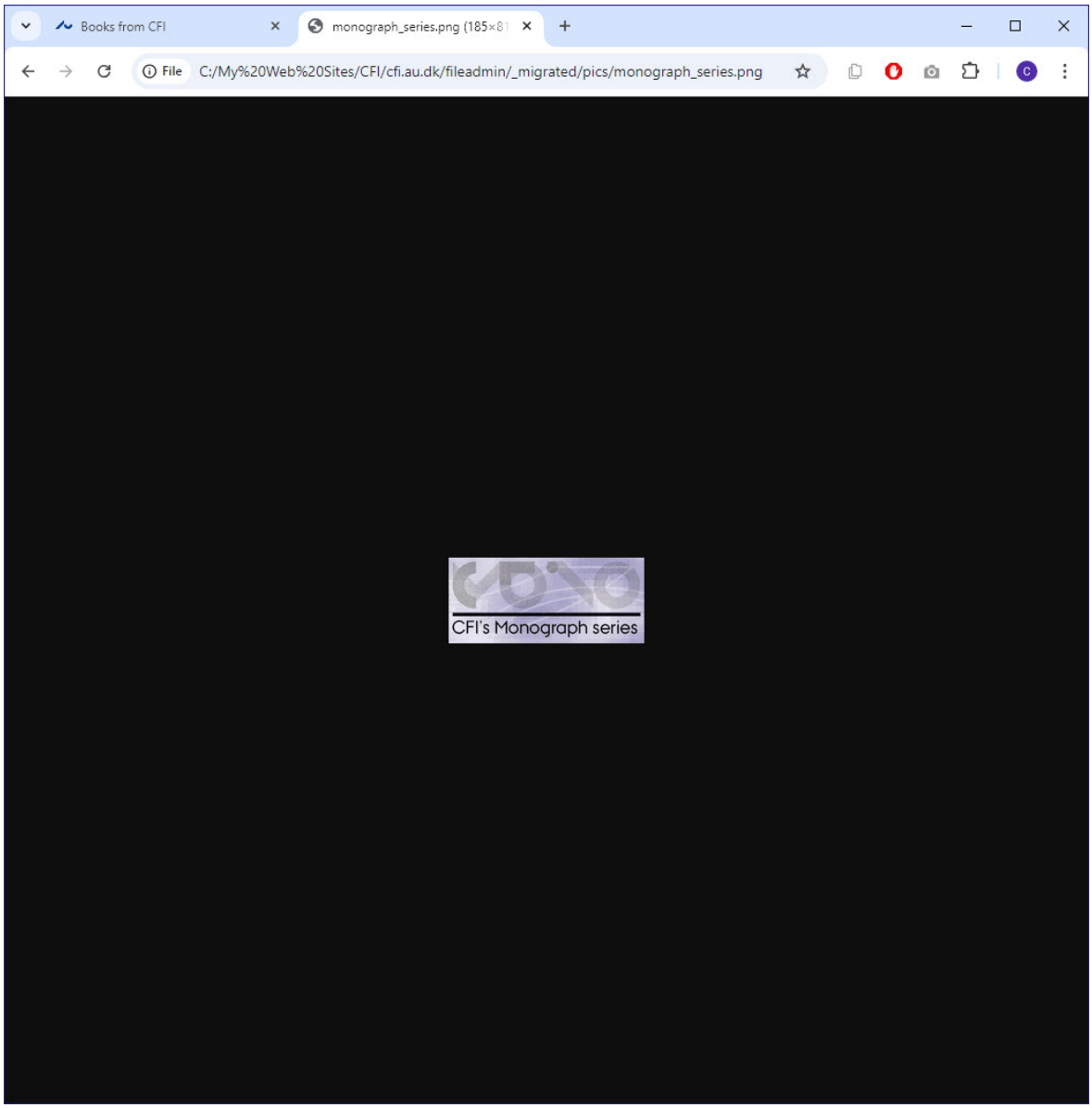


Figure 78: This image is included in the archived copy of the website.

Finding that the image has a local address line at the top of the browser confirms that this is a copy of the image file stored in the local archived copy of the website.

The same happens if checking the cover image for the book announced on the page; that image is also stored. But in the next example the examination proceeds with the Aarhus University logo at the top right of the page:

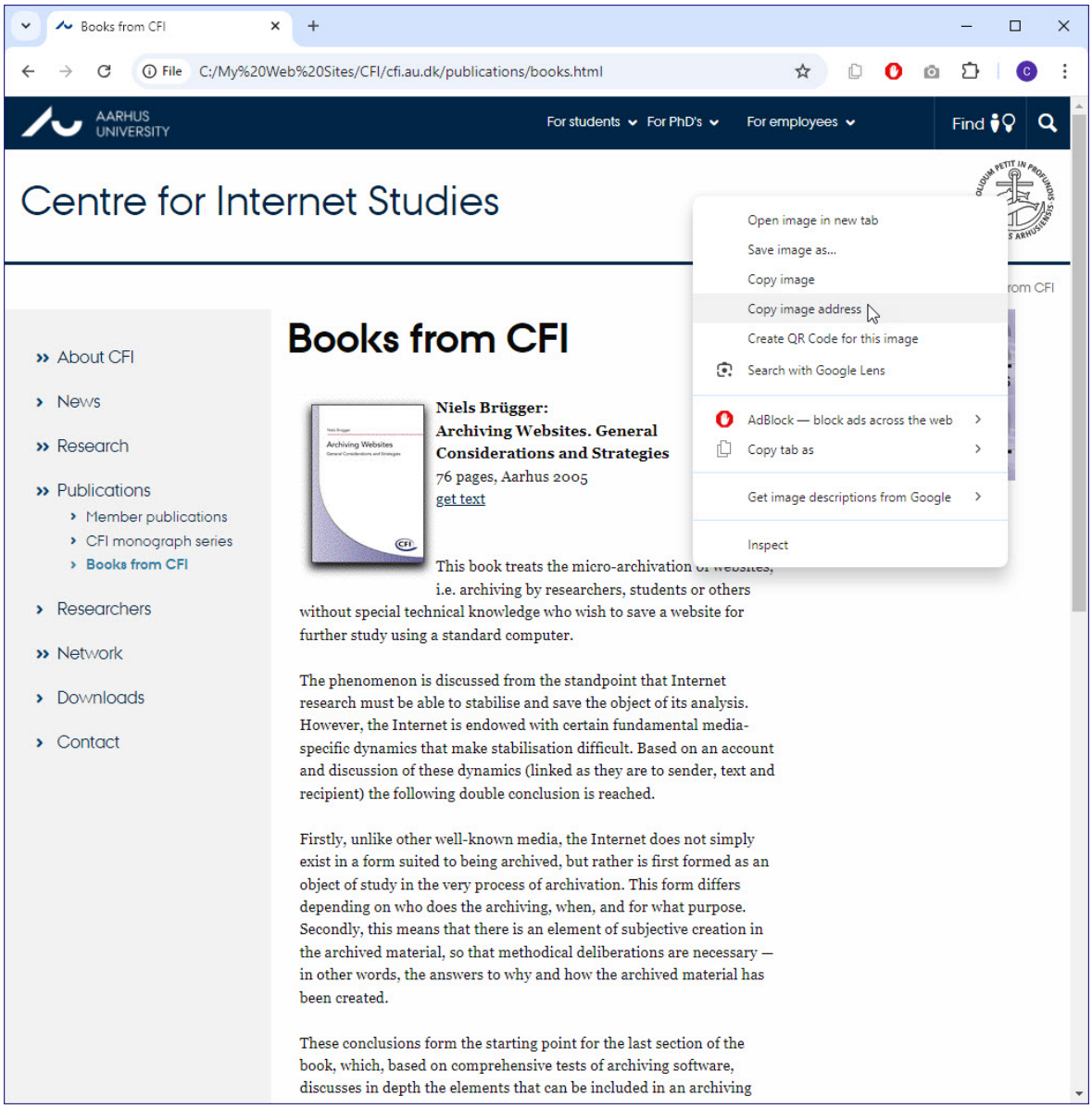


Figure 79: Copying the image address for the Aarhus University logo.

The image address for the Aarhus University logo turns out to be: https://cdn.au.dk/2016/assets/img/au_segl-inv.svg, which is a URL for the live Web and clearly different from a local folder/file address.

Opening that address into a new tab leads to this:

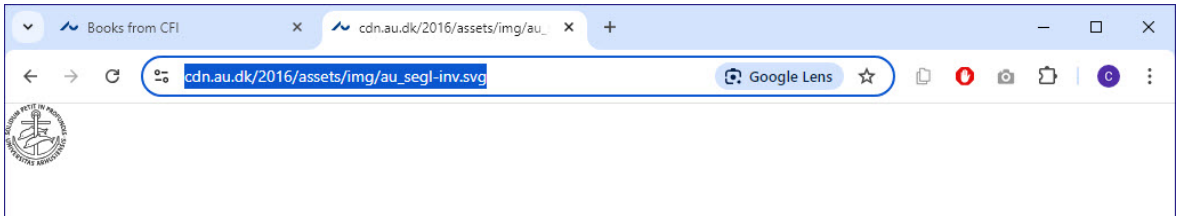


Figure 80: The Aarhus University logo is located online.

What this shows is that the Aarhus University logo is embedded on the page from a location that has not been stored, but exists online.

In other words, the image has been embedded from a resource too far removed from the harvest settings to be retrieved. If the copy of the archived website was to be tested in an offline environment with a cleared cache as described in 7.6.3 Checking Against Online Leaking, Local Archives – then the Aarhus University logo would not be shown. It is only visible in this inspection of the preserved content as a case of online leaking in the archived copy. See also 7.5 The Risk of Online Leaking, and 7.6.3 Checking Against Online Leaking, Local Archives.

In summary, all pages inspected have been preserved, and images directly inserted in support of the content have also been preserved, but there are surrounding elements from Aarhus University's (much) larger website that are not included.

If a very complete copy with all images including the contextual ones from the bigger framework of the entire university website was needed, one would have to do repair crawls with settings allowing for more width and depth as described in 9.4.4 Attempting Repairs.

This might realistically result in a perfect harvest of the CFI website with all retrievable elements preserved, but with large amounts of additional data up to the entire website for Aarhus University and possibly several external websites included as by-harvests.

The conclusion here is that the copy is good, provided that one does not need to be able to show or reconstruct the pages as they were in all details. For that, a much deeper and broader harvest would be necessary, and large amounts of additional data that were theoretically unimportant would be included.

If or when missing elements are encountered, and found to be a problem, it may not be possible to retrieve all elements by doing repair harvests at a larger scale. Various other ways may be called upon in

order to remedy the situation. This depends on the nature of the content, as discussed in the following subchapter.

If one should need to recover the online addresses for locally archived content, the solution is to copy the address for the saved version, and clean away the folder path and the file suffix around it. For example, the address for the local copy of the CFI Publications page discussed above is:

file:///C:/My Web Sites/CFI/**cfi.au.dk/publications**.html

In order to get the original online address for the web page, one must:
1) Delete the folder path leading to the copy, in this case file:///C:/My Web Sites/CFI/, and
2) delete the file suffix since one is looking for a web location, and not a file as such. Therefore .htm or .html will also need to be removed.

Having removed the local folder path and the file suffix one is left with this web address: **cfi.au.dk/publications**. If entered in a browser, this will lead to a live version of the page if it still exists. In an institutional archive it will lead to copies of the page, if the archive has it.

Another type of verification that may also be helpful is to confirm that specific file types have been harvested. For example, there were publications on the CFI website. As with many websites that offer free articles or books for download, the publication format is PDF files.

On can verify that PDF files were included in the archived copy by searching its main folder:

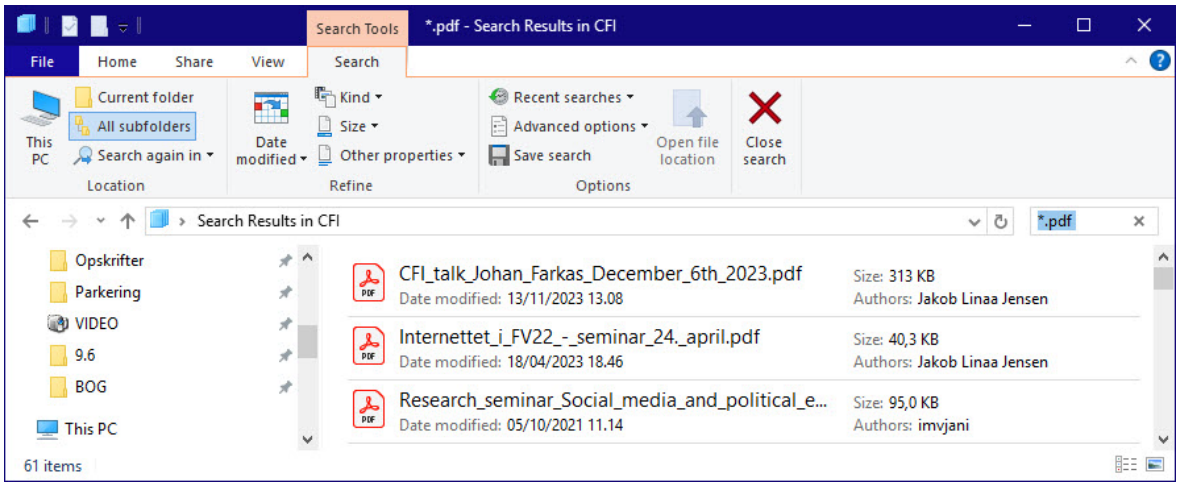


Figure 81: Searching the CFI main folder for PDF files.

Other and similar tests can be conducted for images files, video files, installation files (if applications for download were on the website), etc.

In all these cases there are many different file formats. The reader will either need to know all the relevant file extensions and search for them one by one, e.g. in order to confirm that SVG files such as the logo discussed above were included.

But for this type of test the user will probably do better by knowing, or searching for search operators (e.g. “search for file type mac” to search for operators on Mac OS).

The search operators are the same in Windows and Mac OS, so one can search for “kind:=document”, “kind:=video”, “kind:=image”, “kind:programs” etc. and get an overview of all files of the specified file type in a folder. The advantage is that all types of, e.g. video files such as MP4, MKV, AVI, WEBM, etc., will be found and listed in one search. This has one risk as a side effect; namely that if a specific version of a file type was not harvested it can be difficult to spot that something is missing from the list of harvested files.

Searching for files and file types can be very useful, but a couple of sample verifications by inspecting the harvest results in a browser in order to confirm that they “look good” is the best general way to do a quality check.

9.4.4 Attempting Repairs

If content is missing after attempting to archive a website, it is important to determine whether it is content of a type that it should be possible to harvest, or not.

If the harvest attempt has failed entirely, then the first thing to try is to repeat the attempt with new settings. The location and use of settings will differ from application to application, wherefore one must check the specifications and support sections for the specific software or service. It may be necessary to:

- Ensure that rules in robots.txt are ignored (see 5.1 Challenges for Web Crawlers, p. 77).

- Set the harvester/scraper to mimic a user visiting the web pages in a browser in order to circumvent website programming designed to prevent applications from crawling it.

- Limit the speed of the harvesting process, a) by limiting the bandwidth, or b) by limiting the number of requests or connection attempts. Some websites will block users or applications that send requests too fast, as a protective measure against overloading.

If the harvest has gaps in the form of missing pages, then the necessary step in a repair harvest will be to:

- Allow the crawler to go deeper on the website (depth).

For missing content that should be possible to harvest (e.g. images), then the settings for a repair harvest can include one or more of the following adjustments:

- Verify that missing file types are explicitly listed in the settings for what should be fetched.

- Allow the crawler to go farther away from the target website (external width and depth), e.g. in order to fetch images from a third party host. This can add significant amounts of data amounts and processing time to the harvest, resulting in by-harvests of non-targeted web pages or websites. See 5 The Web Archiving Process, p. 73-77, and 5.1 Challenges for Web Crawlers, p. 82-83.

In web harvesters or scrapers, “repair harvests” can usually be done as additions to earlier harvest attempts, adding the new harvested data to the relevant folders.

For content that cannot be harvested by a web crawler, e.g. interactive content or embedded content from social software, such as a YouTube video – the only option for doing repairs is to document the missing content separately as a supplement to the harvested content.

Depending on the nature of the content, this may call for screenshots, screen recordings, manual copying, dedicated download programs, services for specific content types, etc. As an example, the user could have a folder with harvested websites, and an additional folder with documentation of the missing content, e.g. with YouTube videos that were on some of the original pages.

9.5 Getting Data from Social Media

As described in chapter 6 Social Media, this type of web content is difficult to harvest. Most of the social media services impose strict restrictions on what can and may be harvested, or block harvesting completely.

Please notice:
An important word of warning is that some social media services offer "archiving" or "saving to favourites" for posts. This adds the online and live post that the user may want to be able to revisit, to a personal section of content in the user’s own account. But the content is not saved in a stable form. If a post is deleted, it will also disappear from

the "archive" section of a user's account. If it is changed, the "archive" section will contain the post, but now with the change, not in the form it had when “archived”. A stable copy is a copy that has been taken out and preserved, outside of the API for the social medium, and not a folder or storage area provided by it internally. Adding something to a favourite or “archive” section is similar to adding a bookmark for a web page in a browser; if the bookmarked page disappears, the bookmark will no longer work.

The way that data is handled by, and may sometimes be harvested from social media is described in 6.1 APIs and API Access.

Generally, there are few social media services where data can be saved automatically, and few applications that can do it. The policies for access are handled differently by the companies behind the social media, and if data extraction is to some extent allowed, then major changes in their technologies sometimes make working applications obsolete with short warning.

If one wants to extract data from social media, there are normally services or applications that can extract specific types of data from specific social media, e.g. images or videos. There are for example (presently, in 2024) many applications that can download videos from the social media service, YouTube.

But if one needs to extract posts as data, then this will depend on whether the social medium in question allows some kind of researcher access to their API, or whether one can find a service or software with such access.

Applications that can extract social media data from an API may require some knowledge of programming and/or being prepared to use command lines. This will not be impossible for people with average IT skills, but the learning curve can be steep and is likely to require large amounts of trial and error.

See also 10.8 Software and Services for Social Media.

9.5.1 Saving Manually

If the need for data is not large-scale, manual saving of the content can be done, but the process can easily become time demanding and tedious, e.g. if one would like to preserve an entire discussion thread.

There are some browser add-ons that can automate some of the actions that manual saving would require, but their functionality can be of varying quality and may change swiftly.

In general, when seeing something of interest in social media the user may face three practical problems: Longer posts may not be fully visible until actively opening them in full length, replies or comments may not be visible until opened, and a full page of posts and comments will not be loaded but instead open gradually when scrolling down.

There are basic ways of documenting threads that one has found, but for the above reasons this implies manually clicking on all posts for unfolding their entire content, manually opening lists of comments or replies, and manually scrolling down while repeatedly opening more posts and replies, until the topic ends or loses relevance. There are no universal ways of doing these things automatically, because social media handle their content differently with their respective APIs.

If the need for specific data does not exceed the limit of feasibly opening it all manually, there are a few ways of documenting the content:

- It can be captured in screenshots, at the loss of machine readable text, separate files such as images, URLs for external content – which may not be shown in full and will thus be completely irretrievable – but screenshots can serve as documentation.

- Screen recordings have the same limitations as screenshots, but may reflect the interaction as the user scrolls down and opens

content, which may in this case include images, links, videos, etc. Video screenshots of specific parts may be taken later.

- Specific content such as images or videos may sometimes be extracted with the help of online services, but it may not be possible to automate the process.

- Finally the content can be copied and pasted into documents. The result of this will not capture the look and feel of the content correctly, and in general one can expect to get the text only, while other content such as images will be lost. The primary advantage of this strategy is getting the text in a searchable and machine readable format.

For limited collections of social media content created manually it is advised to combine a couple of these methods, in order to get the most flexible and complete dataset.

9.5.2 API Harvesting

For someone who would like to harvest data from a specific social medium, it is worth searching for information on API access for said medium. Some of the social media services have offered research access to their APIs, but most or all have closed down these possibilities, or replaced them with paid access types.

If one gets access, it is important to read the terms of service, both for regular users of the social medium, and the additional terms that come with special access. The terms of service must be read closely and followed. Failing to do this can result in countermeasures, usually in the form of being blocked temporarily or losing access permanently.

If one has found a social medium that can be harvested, and a software or service that is capable of doing it, then the rules for how the harvested data from the API may be handled must also be consulted and followed, e.g. rules for what may be shared or made public.

Software and services that can harvest data from APIs will extract data from the social medium's database where they are stored in a spreadsheet-like format, as described in 6.1 APIs and API Access. The data can usually be extracted in at least one format which is also recommended for users who want to do research: CSV files. These are basic text files with comma separated values, and can be imported to spreadsheets where the data can be sorted, cleaned, and analysed or prepared for analytical applications.

Other file formats that are sometimes offered are JSON or GDF files.

JSON files are a web data format which is popular for programmers. It contains all the data, often including some technical relations between various data types. JSON files are machine readable, and can also be read by humans in a browser window, but converting them to the CSV format is difficult, and they are also unsuitable for direct use in analytical applications for personal computers. If it is possible to download a JSON version of the data this may be a good idea, especially for complex projects where special programming for the tasks at hand is feasible. JSON files may also serve as additional data at a later point, so having them may be a good idea. But the CSV format is preferable to most users, simply because it is much easier to work with.

GDF files are data files specifically prepared for use in the network analysis program, Gephi. If a project can or will make use of network analysis, the GDF files may be useful. CSV files can also be prepared for such use in spreadsheets, and are attractive because they can be sorted and cleaned manually.

If a project is or may later evolve into a complex one, the broad advice is to go for all options in order to have all data representations at hand.

For most use cases the advice is to focus on CSV files as the data format that is easiest to interpret for humans, and for proceeding to work with the data.

10 Searching for Software and Services

Takeaways

♦ A list of Tools and Tutorials maintained by CDMM and mentioned in the start of this chapter can possibly spare the reader from doing exhaustive searches for software or applications.

♦ If solutions exist for specific data harvesting needs they can almost certainly be found in online searches.

♦ It can be very helpful to search for lists of "best application for...", and for professional and independent reviews of specific software or services.

♦ One may find highly specialised software or code which it can be demanding to get to function, especially if looking for solutions to difficult or complex forms of data collection. Before attempting to use such solutions it is recommended to try to verify if they (still) works for others.

♦ One should be wary of scams, e.g. by searching for information on whether a specific software or service is trustworthy, or for discussions on it, or for free alternatives.

As much as this book attempts to advise on principles rather than specific software or resources that may change soon, it is worth mentioning to the reader that as long as the publisher behind this book exists – Centre for Digital Methods and Media (CDMM) – then a large section on software for various archiving purposes may be found in CDMM's Tools and Tutorials Section, here:

https://cc.au.dk/en/cdmm/tools-and-tutorials

CDMM accepts no liability for use of third party software or services, but that said, the software list at CDMM is tested and the list is maintained so that changes are accounted for; programs that no longer work are removed, and replacements are added if possible, along with new applications or services of interest.

CDMMs Tools and Tutorials list is not a complete list of applications or services, but it is an extensive and maintained list of tested and working solutions for a broad palette of archiving as well as research needs.

When or if the list can no longer be accessed an unmaintained copy can be found by searching for the URL in The Internet Archive, with the possibility of finding references to programs that may still exist, or else serve as keywords to specifically search for alternatives.

But if the need arises for finding software or services for a specific archiving purpose oneself, then the way to do it is by searching the Web. The rest of this chapter will deal broadly with advice for search strategies that may help in the process.

10.1 Considerations Before Searching

When needing to archive some kind of web content for a specific purpose one will often have to search for a solution that may fit the need.

Whether a solution exists, and how to find one (if possible) that is safe, fair priced or free, and not least effective, depends on a number of factors.

While anyone can do a search on the Web, there are steps to help find solutions, select the more promising ones, take precautions against traps, etc., wherefore a detailed description of possible steps may be helpful. The steps are interchangeable, and in most cases only one or a few may be needed.

It must be noted that nothing can be guaranteed. Looking for, selecting, and using software or services is always done at one's own risk.

First of all, some things just cannot be done. But the risk of finding scams that falsely promise to do them anyway certainly exists.

Second, there is a risk of looking for something that was possible at a certain point, e.g. API harvesting of specific social software services that have been open for such actions, but no longer are (see also 6.1 APIs and API Access, p. 91-92). In such cases one is likely to find search results that describe, or offer services, software or code for the purpose in question, but are in fact obsolete and will no longer work.

Third, for archiving purposes that are feasible (e.g. website harvesting) one is very likely to find a plethora of solutions and solution offers, where the challenge lies in determining which ones are the most interesting, promising, or relevant to look further into.

Finally, when a special need must be fulfilled there is no guarantee that even the most suitable software or service will yield results that precisely fit one's purpose.

Therefore the advice in this chapter is given in the hope of being helpful, but without any liability for what may happen in any specific case.

If these remarks sound a bit disquieting then a word of explanation is in order:

First and foremost the question of internet security is a basic and mandatory one for any use of software and services, for any purpose. Common sense is the best safeguard against frauds or malware infections, and this goes for anything from personal computer use or online shopping, to specialised or professional needs.

Even so, nothing in life is ever completely safe. Taking a walk or staying at home also includes small risks, however much common sense is applied to either action. Using good common sense certainly does not imply refraining from any activity at all.

Here are a few examples for the Web and the Internet, where risks may occur that cannot fully be avoided:

It is a known issue that hackers may intrude upon legitimate websites or services and infect their software or web pages with malware, or redirect legitimate links to scam pages. A long-standing trusted website can also be bought or hijacked by less honourable parties that may rapidly or gradually change the site to something less secure or desirable.

Ad banners on legitimate websites have also been used for criminal or fraudulent purposes. And there are lots of other types of fraud such as scam sites that look genuine and are named so they may easily be confused with legitimate websites, scam websites that boast good reviews which are actually false, etc., etc.

The good news is that most of the time the user is well protected by observing the basics of internet security: Keeping their software and operating systems updated, having security programs such as antivirus and firewalls well in place, using strong passwords, etc.

Even so, trusted and reliable services are hacked from time to time, resulting in theft of private and personal data, and even identity theft.

The point is that common sense and recommended security measures are the best safeguards against fraud and crime, but never a full guarantee, neither on the Web nor anywhere else.

The guidelines and recommendations laid out in this chapter are meant so support one's chances of finding results and steer clear of the more obvious risks in the process.

10.2 Search Strategies

When looking for software or services for a specific purpose, it is usually worth doing a couple of broad searches. Different search criteria and approaches may help one to narrow down the most promising solutions. One reason for this is, that the more promising solutions will most likely appear in most or all relevant searches.

Depending on what one needs, some simple functions may be obtained with browser extensions or add-ons. This is relevant if one specifically has the thought "I wish I could do X right now" while browsing, e.g. save a good copy of a page, extract a background image that cannot be saved by right-clicking and choosing "save picture", save all URLs from many tabs in one click, download a video from a page, etc.

In such cases, one's search should start in the catalogue of browser extensions, and possibly not be limited to the catalogue for just one browser. Please see 10.6 Browser extensions for further details on this.

But for systematic archiving, more complex functions will usually be relevant.

Let us say for example, that one were to look for a program or a service that can save an entire website, with the limitations described in 5.1 Challenges for Web Crawlers and addressed in 9.4.3 Checking the Quality of a Harvest.

If one was without advance knowledge that such a thing is possible, or unaware that the terminology for such programs or services would be expressions such as “web harvester” or “web scraper”, then the search strategy should be to attempt queries on what one needs to do. This could be phrased in queries such as:

“software for saving entire website”

“best software for saving entire website”

“best free software for saving entire website”

“service for saving entire website”

“best service for saving entire website”

“best free service for saving entire website”

The queries can be given in any order, and one will probably not have to do more than one or a couple of them. On the other hand, depending on how many, or few, or confusing results one gets in a specific search, queries may also be varied or expanded upon.

One may try adding the present year and see if this has an impact on the top hits. This may rule out articles that would otherwise be helpful, wherefore a search without year should also be tried, but adding the year can be helpful for finding solutions that are presently relevant in cases where technologies and possibilities have changed.

If one wants to have the best options possible for a task, it may also be relevant to search for “best professional program for” and “best professional service for”.

If one is fairly certain that what their needs are technically possible, then a search for “best software for” will often suffice. It will often lead to lists or articles which discuss or compare various solutions, free as well as paid.

10.3 Types of Queries and Results

All the types of queries mentioned above may lead do different results, from very relevant direct hits to highly technical discussions. They do however shuffle the treatment of the primary topic by shifting the focus.

The "can it be done" approach can often yield a couple of hits for discussions fora and/or professional articles.

If a discussion found in a forum, or an article from a website for computer users/enthusiasts/specialists, is a couple of years old or more, such results may well be worth taking a closer look at, but this could also be a reason for doing an extra search with the present year included as a search term, in order to see if newer advice is readily available.

Discussion fora:
People with a need very similar to one's own have very likely asked others to help with possible solutions, often but not always in a forum with a technical specialisation. One therefore has a chance of finding a number of helpful replies describing methods, or recommending specific software for a task. The disadvantage is, that such discussions will only list the solutions known to the people replying, and may reflect very different approaches or technical skill sets. Recommendations will be discussion submissions in a random order, as opposed to a ranked list.

Reading or skimming through discussions may help one to a) find relevant mentions of software or services worth looking further into, and b) benefit from technical explanations and comments by getting a better idea of the challenges or ease for the task at hand, and advantages/disadvantages of different approaches.

Professional articles:
This type of articles will often be written by computer experts at computer websites or magazines. The great advantage to such articles is that a lack of commercial bias and a solid amount of technical insight can be relied upon. Such articles are still written by human

beings with their own personal preferences, and if more are found by different authors it is quite common to find many of the same services or programs listed, but with slightly different rankings. If one author ranks software X as the best choice, another author may consider it only second best or less, and rank software Y as the best solution.

Still, this type of articles can have the great advantage of providing insights. If the task at hand is difficult, and only one or few solutions exist, professional articles will explain why this is the case, and often explain in detail how the viable solutions may be brought to work. The rankings one may find are less important; more important are the pros and cons specified for various products, since they may help one to find the solutions that are most interesting for further pursuit, comparison, or investigation.

A listing for e.g. “The 10 Best Free Solutions for X” is not likely to cover all possible solutions, and there may be others that are as good, or better. It is fully possible that one or two such articles may lead to a fully satisfactory solution. But the more specific one’s needs, and the less that fully compatible solutions are found, the better reason there will be to also look for technical discussions with the “can it be done” approach.

One thing to be wary of for the professional type of articles: In some cases one may find articles that appear typical of the genre, with comparisons between various solutions – but which are written or published by a specific company and usually arguing that said company’s product is the more recommendable. It will hopefully be fully clear and visible that the article is hosted on the website for the specific company in question. Such articles are not dishonest per se; their arguments may be valid, and competing products may be loyally reviewed and free alternatives may be mentioned – but even so some bias can be suspected.

The “best software” and “best service” approaches have the advantage of typically finding direct hits to articles discussing the pros

and cons of a number of solutions that are tried and tested, and likely to represent a number of the better choices that one may have.

The main reason to do these types of searches is that they may often provide good and direct solutions. The difference between “best software” and “best service” is more complex:

“Best software” may provide listings of applications that one will need to install, and which may vary highly in complexity and ease of use, as well as pricing, wherefore an additional search for “best *free* software” is also recommended for comparison. The advantages of finding software solutions are primarily that the user will control the process, and also the data if the program stores them locally. It is also more likely to find free software than free online services, although the latter may also exist, or even be plentiful.

“Best service” queries can lead to online solutions where data requests are handled by software run by a service provider, but it can also lead to software applications since they also provide or conduct a service. If the data is handled by a service provider, this can result in data that can be downloaded directly. In such cases the data may be stored in a user account, or sent to a cloud solution where the user has an account, or it may only be available temporarily and disappear after closing a session. In all cases this raises a question of a possible security issue with how well the data is protected, both on the service providers’ side, and on the users’ online security in the form of secure passwords, double verification, data encryption, how secure the user’s own storage solution is overall, etc.

Therefore it must be taken into account whether services where data is stored online or in cloud solutions are compatible with a) the legal framework, if the data is in any way sensitive, and b) local policies at one’s workplace.

Services hosted online may in some cases be the only viable solutions for a task. This will be the case if the solution works with a highly

specialised software solution run by the developer, and is not offered (or too capacity demanding) for local and personal computer use.

Such online services can be paid or free. If mostly paid solutions are in the hits for a general search, a specific search for “best free online service” is recommended, simply to see if such solutions exist.

10.4 Beware of Scams

If one has found promising applications or services in more articles or forums, there should be no problems or concerns with them.

But if a search result is found for an application or service that is not reviewed elsewhere, or boasts positive reviews that cannot be verified, or that promises something that other search results suggest is not realistic or possible, then the offer may not be genuine.

The service or developer may be new and legitimate, but some testing should be attempted. With a lack of reviews, contacting the service provider may not be the best idea since it will verify nothing.

The advice here is to search broadly for customer reviews in services such as Trustpilot, but also to do searches such as:

“is [domain.topdomain for service X] a scam”

This will result in hits with various scam testing services that check the provenance of a website and report signs of scamming, e.g. “the website is opened very recently”, and “has few, recent, and positive reviews” – or signs that it is unlikely to be a scam, such as, “the website has existed for a couple of years”, or “has few but mostly positive reviews”. Recent positive reviews for a new website may be false reviews, while positive reviews over time are a good sign.

10.5 Search for Alternatives

If one knows of an application or service that was once good for a specific purpose, but which no longer works or exists, it is often possible to find alternatives simply by searching specifically for that.

If searches have resulted in one or a few promising solutions, then these can also be used in a search for alternatives, if one would like to attempt broadening the field of possibilities.

A useful query can be:

“alternative to [service or application name]”

When searching for a replacement program or service, consider doing so both with and without present year included in the search terms.

Other iterations of search terms, such as “best” or “free” may also apply here.

10.6 Browser Extensions

Most modern browsers have the option of adding browser extensions, or add-ons, which will provide new functions for the browser.

The best way of finding relevant add-ons are by searching the official catalogue of extensions for the specific browser. The way to do this may vary slightly.

In most browsers there is a settings or costumisation menu at the top right corner, usually identified with either three small dots vertically; ⋮ (popularly known as a “kebab menu”), or three lines horisontally; ≡ (known as a “hamburger menu”).

The settings menu is useful for many things. From here, one can change the preferred search engine, start page(s), set the browser to start with the windows that were open in the last session, find and manage passwords, and manage many other browser behaviours.

When entering the browser settings menu, a section for extensions/add-ons should appear. From there one has the options of managing already installed add-ons, or search for add-ons. In some browsers these options are given on the same page, in others management of installed add-ons and searching for new add-ons are divided into in separate submenus. For example, in Google Chrome one must (presently, 2024) choose between “Manage Extensions”, or “Visit Chrome Web Store”. The latter is similar to app stores for computers, cell phones or other mobile devices.

If in doubt, the reader will have to search for how to find add-ons for a specific browser, e.g. by searching for “find and install add-ons for [browser name]”.

If one tries to search for a specific function, several offers of add-ons may be found. If nothing is found, try other search terms, and if there are still no results, try in another browser – where the function one needs may be available. With different programs and program architectures (“builds”), some functions may be easily added to one browser, and difficult or impossible to add to another. Of course, if nothing is found anywhere, the specific function may not be available, but it will never hurt to look.

If extensions are found, take a closer look. An extension may have only a few reviews but many users. The larger the number of users, the better the chance that the extension works well, and does not cause issues. An extension may be too new or too specialised to have a large number of users, in which case looking at reviews or searching for users commenting on it elsewhere may also be helpful.

At the page for a specific extension there are usually closer descriptions of how to use the extension and how to use different function or settings, if applicable. This can be helpful for choosing the best extension for one’s purposes.

Most extensions are free of charge, some require some form of payment for an add-on support program, or a subscription. In this case

look for free alternatives, and compare with what one will get in the paid version, as well as how other users comment on whether the paid version delivers extra value for the money.

As an example, imagine that one is searching for websites of a specific type, and would like to proceed with further work, e.g. harvesting the pages, or just saving their URL addresses as references for future use. Various relevant pages have been opened in new tabs, and possibly some less relevant have been closed again. One now has a window with many tabs with good search results. Now one just needs to save the URLs. But in a normal browser this means that one will have to manually copy each address, one by one, and insert them elsewhere, probably in some kind of document.

If one searches for proper function descriptions, e.g. “save tabs”, “save URL”, “save tab URLs”, or iterations with “copy” instead of “save”, then one is very likely to find an add-on that can copy all the URLs from a browser window in 1-3 clicks.

Readers who try this may not only find such an extension useful; they may also find that different add-ons for the same purpose act in ways that are more or less useful for different purposes. An add-on may do just what was intended and allow copying of all URLs from a browser window – or it may add the page titles above each URL, or other types of information such as time stamps. Versions with added information may be useful for remembering and managing one’s results, but the pure list of URLs is more useful as a seed list for harvesting the websites. It may thus be worth checking more add-ons, and perhaps to have different functions to choose from, possibly in different browsers or by finding add-ons that can be configured for different behaviours.

Readers who might want to pursue the idea of copying many URLs from the same windows automatically should also read 9.3.1 Tracking Information in URLs.

Other examples of useful add-on functionalities could be, e.g., save better copies of web pages, extract background images that cannot be

saved by right-clicking and choosing “save picture”, download a video from a web page, etc. These examples were also mentioned in the subchapters 9.2.1 Saving Directly from the Browser, and 10.6 Browser Extensions.

It is usually recommended to get add-ons from the official app stores for the different browsers, but this is no guarantee of getting a useful or non-malicious app. The best strategy for that is to pay attention to the number of users, their comments, and whether the add-on has been used to other people’s satisfaction for some time.

Similarly, it is possible to find good add-ons externally on home pages or in developer communities, but then one will still have to try to determine the trustworthiness of what is offered, and by whom.

Important: The reader should notice that browser extensions usually come as free services developed by one or more programmers for a specific browser at a specific time. When a browser is changed fundamentally by major updates – this occurs less frequently than regular updates, but it does happen occasionally – this may cause some add-ons to stop working. In this case, the developers behind an add-on may provide a new version for the changed browser after a relatively short time. But it can also happen that the developers are no longer active, or that the new version of the browser makes it impossible to provide the specific function(s) anymore. If an extension stops working, the best thing to do in that case is to look for a replacement, possibly in another browser. If an extension exists for more browsers, there is also a chance that another browser will still support it.

10.7 Scripts and Command Line Interfaces

When searching for software or services that “can do something” it may occur that one finds references to code or special programs that requires installation in a software environment dedicated to running code in a specific programming language, e.g. “Python” or “R”.

This is the case especially for advanced functions for web or social software monitoring or “scraping”, or explicitly for API harvesting from Social media.

When or if such dedicated code works well, it can be the best way to get automated harvests of data for sources that are otherwise difficult to work with.

If something like this is encountered, first of all check the age of the findings. If they are old, i.e. dated a couple of years ago, there is a risk that they were built for something specific that could be done at the time, but no longer.

Also, search for the name of the function with search terms like,
“does function X still work” (where X is the most precise reference possible to the function in question, such as function, script, developer, e.g. “does Zarribag's script for API harvesting from Twitter still work”), “alternatives to function X”, or “replacements for function X” which will almost certainly result in hits if “function X” has ceased to work.

If it looks as if the function in question is still working and maintained by active developers, and if there does not seem to be an easier way to do what one needs, then it can be worth going through the process of testing it and attempting to get it to work.

This can be difficult, but one can keep in mind that others can make this work (and search for advice from those that may have shared it).

Here are a few points to consider in the process:

- Check for all possible specifications and instructions.
  Such specifications may include installation directions, prerequisites such as first installing a software environment, lists of commands, etc.

- Developers that are kindly sharing their work are used to working with the specific type of code and the software environment that

may be required, and they may expect the same to be the case for potential users. If the specifications one can find for a specific function do not seem to give sufficient directions, it may be necessary to do a broader search for “how to install and run scripts in language X”. Some basic learning in that area may be required before one can proceed.

- Be careful to follow all steps precisely and in their correct order. Pay attention to details such as whether the computer’s “terminal” or “command console” should be used for entering commands, or whether commands should be entered elsewhere, e.g. in a prerequisite software environment.

- Follow directions with extreme precision. If downloaded files need to be placed manually in a directory deep in the computer, then that directory must be used. If a program must be installed by entering a command line, then the command line must be entered precisely.

- Command lines have a syntax – a precise formula for how it should be built and the expressions that can be used. If one attempts to run a command with the slightest syntax error, e.g. forgetting a space after a “-“, or using a lower case letter when an upper case letter is expected, then the command will not work.

Some trial and error must be expected, even if following instructions carefully.

It may be necessary to search specifically for what fails, e.g. by searching for “cannot get Command X to work in Function Y”, or to review or retry the entire process step by step.

10.8 Software and Services for Social Media

For social media there are usually applications or services that can extract video or images. If this is what one needs, that is what to look for in broad searches.

For data harvesting of such things as posts, threads or comments, the best thing to do is the get API access for the relevant service if possible, as discussed in 6.1 APIs and API Access. For this, one should start with searching for:

“API access [service name]”

If API access is offered in any official form, paid or free, then the search will lead to hits at the service provider’s own domain. There, the terms of getting access, the levels of use that will be possible, and the conditions one must adhere to will be stated clearly and systematically.

If API access is offered – and obtained – the next thing to do is to find out how to harvest the data one needs. It is probable that recommended software is listed by the service provider in connection with other information on API access, and even that instructions for use are given.

Most types of programs that can harvest from a social medium’s API will be of the types mentioned in the previous subchapter, 10.7 Scripts and Command Line Interfaces.

If API access cannot be obtained, the next best thing is if there are programs or services that can extract some or all of the data one needs. Finding such services, when or if possible, depends on search procedures and strategies laid out in this entire chapter.

11 Legal and Ethical Concerns

Takeaways

♦ Laws and regulations on person sensitive data and copyright have a large impact what may legally be done with harvesting, storing, sharing, and (re-)representing data.

♦ One becomes data responsible when harvesting and storing data, unless one works in a department where a specific person has the overall data responsibility, in which case one is still responsible for adhering to regulations as set up by their workplace.

♦ Researchers or students at universities may find that the policies and regulations of their own institutions serve as a framework which is defined to keep data handling within the legal regulations.

♦ Besides from legal considerations, ethical concerns are also relevant in respect of web data and archived web data. It is recommended to consult ethical guidelines for the discipline(s) most closely related to a specific study.

Legal and ethical concerns for archived web and web archiving are a highly complex issue, and there is no kind of “one solution fits all” for it. Different countries and states have different laws on data protection, data rights, and protection of sensitive data, especially person sensitive data that can be traced back to an individual.

If one is a researcher or a librarian/curator working with web archiving, there are special legal concerns which in most states or countries will acknowledge that one’s work is in the service of societal and common interest, allowing for a larger extent of freedom to gather and handle data in the course of one’s professional work than will be applicable to most other individuals.

Researchers and students at universities, and other persons that work with web archiving in a professional and institutional context will be best served with finding and adhering to the rules and guidelines of one’s institution. This can spare one from large amounts of frustration, because the institution’s rules and guidelines are defined to protect against law-breaking or clearly unethical conduct. By adhering to institutional rules, the student or researchers will stay within legal and ethical boundaries that their institution is responsible for.[31]

Those who are not covered by following an institutional framework and policy face much more severe problems. They will have to:

- Observe copyright rules, which allow quoting but not to the extent of reproduction. Relevant rules depend on legal use as defined by the copyright holder, and the type and level of copyright, i.e. traditional public copyright licenses versus Creative Commons licenses or Open Access.

[31] Readers that are not affiliated with or enrolled at a university may also want to examine the framework for legal and ethical data handling at their most nearby universities. The attention should be directed at the guidelines for students, since the researchers’ right to do work in the common interest of society will not apply. Guidelines for students may not apply completely either, but they may potentially be a shortcut to finding the most important ethical considerations and legal points that can serve as reasonable boundaries for such readers’ own conduct.

- Follow their state or nation's laws pertaining to data access, handling and storage.

- Follow the laws of any kind of federation or union that their state or country is part of.

- In many cases follow rules that are enforced by other federations or unions.

As an example, the legal framework in Denmark is highly protective in respect of person sensitive data. Since any collection of websites on any topic may include content that a named or traceable person wrote a number of years ago, any data found or extracted in the Danish web archive, Netarkivet, is potentially person sensitive. That is why, due to the legal framework, the national Danish web archive may only provide access to researchers, since their work is covered by an exception in being by definition "work in the common interest of society".

As another example there is the European Union's General Data Protection Regulation. The full name of the last version of said regulation is, "Consolidated text: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC".

This 78 pages long legal text is more commonly know by the abbreviations "General Data Protection Regulation", or "GDPR".

While the regulation is created by the European Union, and applies to the Union's citizens, it has international implications since it protects EU citizens from any kind of abuse of person sensitive data as defined by the regulation. Someone who lives in any other place than the EU may therefore still risk legal repercussions if violating the GDPR regulation.

In short; if one is not a researcher, a student, an archivist, or a librarian with a dedicated legal framework for working with and handling person sensitive data, there is hard work ahead with finding the relevant legal framework for working with Web data and staying within legal boundaries. Not only publication, but also acquisition, possession, or sharing of data that contain person sensitive information may be illegal.

It is up to the individual reader to determine whether the reaction to all this should mostly align with the tagline for Douglas Adams' novel "The Hitchhiker's Guide to the Galaxy" (1979), "Don't panic", or with the tagline for David Cronenberg's movie "The Fly" (1986), "Be afraid. Be very afraid." But in most cases the proper reaction will probably be somewhere around the middle between the two, because:

After all, the legal framework consists of laws that were designed with the intention of protecting copyright owners and the privacy of individuals, and not with an intention of being a hindrance for research.

11.1 Terms of Service

Terms of Service (commonly abbreviated as "ToS") is a widespread expression that may cover the use of services provided on a website, e.g. with forums and user registration, or the agreement between an Internet Service Provider and a person, or the use of services offered by social media.

In some cases the ToS is not legally binding. For example, if a publically accessible (i.e. not password-demanding) website states that "you may use the content this way or that, but not in such a manner as...", this may or may not be legally binding, depending on the nature of the content and service provided, and the restrictions stated. The possible repercussions of breaking such a ToS can be anything from angering the website owners (if they find out) and to breaking copyright laws.

In other cases a ToS is legally binding in the sense, that if someone breaks the ToS the owners of a service have a legal right to stop the use of their services, e.g. by closing an Internet connection, and possibly reporting illegal activity depending on the severity and type of violation.

If working with social media, their ToS must be observed in addition to other legal concerns and frameworks. This is because the Social Media providers are privately owned companies, that determine their own sets of rules for using the service, as well as the extent to which data may be extracted and how it may be used. Breaking the ToS of a privately owned company is basically the same as breaking any other legally binding agreement or contract.

Both the general Terms of Service that apply to users, and the additional ToS for API access if applicable, must be observed by anyone using a social service. These rules typically determine ownership and codes of conduct. Failing to observe them can result in the user's account being temporarily or permanently banned and/or deleted. Failing in a large-scale or gross manner can result in lawsuits.

It is therefore extremely important to close read, and adhere to, the ToS for a social media service if one wants to use their data for research. Simply put, this adds an extra layer of legal concern in such a context.

11.1.1 Ethical Framework

When using data for research with the implication of giving examples, or making the data accessible to a wider audience, ethical concerns are called for in addition to legal concerns.

For researchers and students, their universities will normally have a list of guidelines in this area, too, and following these is a good and secure start.

But depending on the nature of one’s work, e.g. mostly philological, mostly social science, mostly media studies, mostly anthropological,

and depending on the topics of the project and its data, e.g. mostly public debate, mostly art, mostly official political statements, some additional concerns may be relevant.

For example, does the nature of the data call for blurring of images, or specifically of faces, or hiding the names or aliases of persons?

One may search for guidelines for ethical conduct in the relevant field of study, specifically outlined by and for research communities in that field.

There will often be more research communities for a field, and their guidelines may differ on specific questions.

For example one may search for “ethical guidelines [name of field of study]”, and find guidelines as laid out by various research communities such as;

Field of study:	**Ethical guidelines found from, e.g.:**
Internet research	Association of Internet Researchers (AoIR)
Communication	The International Communication Association (ICA)
Social science	Social Research Association (SRA)
Anthropology	American Anthropological Association (AAA)

Finding and reading relevant ethical guidelines can lead to informed choices for how specific types of data will be handled and represented, with solid ethical arguments to back them up.

12 Further Reading

If the reader should wish to read more, the following sources are recommended:

Niels Brügger (2017): The Web as History,
UCL Press, available as a hardcover, or as an open access e-book,
https://uclpress.co.uk/book/the-web-as-history/

Niels Brügger (2018): The Archived Web - Doing History in the Digital Age, The MIT Press: Cambridge, MA,
https://mitpress.mit.edu/9780262549714/the-archived-web/
Free digital edition, ISBN electronic: 9780262350112 available on:
https://direct.mit.edu/books/monograph/4215/The-Archived-WebDoing-History-in-the-Digital-Age (Registration required)

Niels Brügger, Ian Milligan (Eds.) (2018): The SAGE Handbook of Web History. London: SAGE
https://uk.sagepub.com/en-gb/eur/the-sage-handbook-of-web-history/book252251

Internet Histories (2017 and forward), Scientific Journal, Edited by Niels Brügger, Anat Ben-David, Ian Milligan, Valérie Schafer, and Emily Maemura (book reviews), with occasional guest editors. Published by Taylor & Francis
https://www.tandfonline.com/journals/rint20
(The author of the book you are presently reading is said journal's editorial assistant wherefore this recommendation cannot claim full neutrality, but the journal does cover a very broad range of different uses and studies of archived web).

References

Explanatory note:
Author names are given in full with first name and last name, e.g. “Janne Nielsen” rather than “Nielsen, J.”. References are sorted alphabetically by first author’s last name and after that chronologically.

Online references are mostly given as Internet Archive URLs. Excepted from this are articles where the full version must be purchased and will therefore not be accessible in an open archive, and scientific resources where explicit requests are made for DOI citations[32], or where a free version is offered. Those references simply give the URL; if it ceases to work then the URL can be found in The Internet Archive. Please refer to 8.3 Referencing from Web Archives for background on how to refer to Web resources.

In order to retrieve online versions (if still available) from archive URLs, simply use the original web address contained in the archive URL after timestamp/.

E.g. remove https://web.archive.org/web/20240531035343/ from the address
https://web.archive.org/web/20240531035343/https://ieeexplore.ieee.org/document/6970226/.
The original URL or web address can then be used to check for a “live” copy, in this case: https://ieeexplore.ieee.org/document/6970226/.

- - - - - - -

[32] DOI stands for "Digital Object Identifier"; a standardised type of URL in the form of a string of numbers, letters and symbols used to identify the location for an article or an object on the Web.

Teru Agata, Yosuke Miyata, Emi Ishita, Atsushi Ikeuchi & Shuichi Ueda (2014): Life span of web pages: A survey of 10 million pages collected in 2001, IEEE 2014, https://web.archive.org/web/20240531035343/https://ieeexplore.ieee.org/document/6970226/

Mohamed Aturban, Michael L. Nelson & Michele C. Weigle (2021): Where Did the Web Archive Go?, arXiv:2108.05939, https://web.archive.org/web/20240928113951/https://arxiv.org/abs/2108.05939

Niels Brügger (2005): Archiving Websites. Centre for Internet Research, 2005, https://web.archive.org/web/20240826150727/https://cfi.au.dk/fileadmin/www.cfi.au.dk/publikationer/archiving_underside/guide.pdf

Niels Brügger, (2018): The Archived Web: Doing History in the Digital Age, MIT Press: Cambridge, MA, 2018. Free digital edition, ISBN electronic: 9780262350112 available on https://direct.mit.edu/books/monograph/4215/The-Archived-WebDoing-History-in-the-Digital-Age

Niels Brügger (2021): Digital humanities and web archives: Possible new paths for combining datasets. International Journal of Digital Humanities 2(1), p. 145-168, https://doi.org/10.1007/s42803-021-00038-z

Niels Brügger, Ditte Laursen and Janne Nielsen (2017): Exploring the domain names of the Danish web, *in* Niels Brügger and Ralph Schroeder (editors): The Web as History, UCL Press 2017, https://www.dbooks.org/the-web-as-history-1911307568/

Laura Ceci, Apr 11 (2024): Hours of video uploaded to YouTube every minute as of February 2022; https://web.archive.org/web/20240416030557/https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

The European Union (2016): Consolidated text: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation, European Union Law, 2016). Latest version used in this publication: https://web.archive.org/web/20240823150950/https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504, for updated versions if relevant check references to updates at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504

Chris Freeland (2024): Internet Archive and the Wayback Machine under DDoS cyber-attack, https://web.archive.org/web/20240729060655/https://blog.archive.org/2024/05/28/internet-archive-and-the-wayback-machine-under-ddos-cyber-attack/

Mark Graham (2017a): Robots.txt meant for search engines don't work well for web archives, https://web.archive.org/web/20240729224928/https://blog.archive.org/2017/04/17/robots-txt-meant-for-search-engines-dont-work-well-for-web-archives/

Mark Graham (2017b): Wayback Machine Playback… now with Timestamps!, https://web.archive.org/web/20240824165729/https://blog.archive.org/2017/10/05/wayback-machine-playback-now-with-timestamps/

Andy Jackson (2015): Ten years of the UK web archive: What have we saved?, paper presented at the 2016 IIPC GA, Palo Alto, https://web.archive.org/web/20170315185608/http://netpreserve.org/sites/default/files/attachments/2015_IIPC-GA_Slides_03_Jackson.pptx; https://web.archive.org/web/20240302140838/https://blogs.bl.uk/webarchive/2015/09/ten-years-of-the-uk-web-archive-what-have-we-saved.html

Victor Mayer-Schönberger & Kenneth Cukier (2013): Big Data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt Publishing Company, New York, 2013

Janne Nielsen (2016): Using Web Archives in Research, NetLab 2016, https://web.archive.org/web/20220312120220/http://www.netlab.dk/wp-content/uploads/2016/10/Nielsen_Using_Web_Archives_in_Research.pdf

OECD (2016), Skills Matter: Further Results from the Survey of Adult Skills, OECD Skills Studies, OECD Publishing, Paris, http://dx.doi.org/10.1787/9789264258051-en; https://web.archive.org/web/20160702141108/https://www.oecd-ilibrary.org/education/skills-matter_9789264258051-en

Jack Pembroke-Birss (2021): Using screenshots from The Wayback Machine in court proceedings, https://web.archive.org/web/20220503031739/https://nortonrosefulbright.com/en/knowledge/publications/57e50249/using-screenshots-from-the-wayback-machine-in-court-proceedings

Alexis Rossi (2017): If You See Something, Save Something – 6 Ways to Save Pages In the Wayback Machine; Internet Archive Blogs, https://web.archive.org/web/20240731135122/https://blog.archive.org/2017/01/25/see-something-save-something/

Valérie Schafer & Ben Els (2020): Exploring special web archive collections related to COVID-19: The case of the BnL, Warcnet Papers and Special Reports, https://web.archive.org/web/20230902214738/https://cc.au.dk/fileadmin/user_upload/WARCnet/Schafer_et_al_Exploring_special_web_archives.pdf

Eveline Vlassenroot, Sally Chambers, Sven Lieber, Alejandra Michel, Friedel Geeraert, Jessica Pranger, Julie Birkholz & Peter Mechant (2021): Web-archiving and social media: an exploratory analysis, International Journal of Digital Humanities (2021), Issue 2, https://doi.org/10.1007/s42803-021-00036-1; https://web.archive.org/web/20240415095835/https://link.springer.com/article/10.1007/s42803-021-00036-1

Michele Weigle (2024): Some URLs Are Immortal, Most Are Ephemeral, https://web.archive.org/web/20240923170001/https://ws-dl.blogspot.com/2024/09/2024-09-20-some-urls-are-immortal-most.html

Jane Winters (2017): Breaking in to the mainstream: demonstrating the value of internet (and web) histories, Internet Histories, 1:1-2, 173-179; http://dx.doi.org/10.1080/24701475.2017.1305713

Eld Zierau (2022): PWID Poster, Netarkivet; https://web.archive.org/web/20240810105403/http://id.kb.dk/pwid/PWID.ppsm

Eld Zierau (2022): URN Namespace Registration for Persistent Web IDentifiers (PWID); https://web.archive.org/web/20240627104056/https://www.iana.org/assignments/urn-formal/pwid

List of Figures

About the author

Asger Harlung, e-Learning Consultant, MA, Digital Methods Consultant

- has been a factory worker, a programmer, an adults' teacher, an assistant university teacher, and an e-Learning consultant at the National Centre for e-Learning in Denmark. Since 2016 he has worked with web archives and web archiving and taught this topic at workshops and courses for PhD students and researchers. He is presently working in the broader field of digital methods at Aarhus University, Department of Media and Journalism Studies, and Centre for Digital Methods and Media.

About this book

Web archives are repositories of content preserved from the World Wide Web, not to be confused with other types of archives that offer services or content online.

Web archives preserve content that otherwise tends to disappear or change rapidly, which probably makes them the most important initiative towards preservation of cultural heritage in our age.

Never before have the actions, ongoings, sentiments, trends, decision processes; in politics, business and industry, institutions, news media, entertainment providers, interest groups, and individuals, been documented or preserved at the level of detail offered by ongoing and systematic preservation of the Web, with content from all levels of society.

This book strives to present the underlying methods and reasons for preserving web content, with a focus on the principles (rather than specific methods that may become obsolete almost as quickly as web pages may disappear), and on detailing the nature of the archived results with its potentials and pitfalls.