Digitale indgangsvinkler til webhistorie

Af: Victor Harbo Olesen

GLAM workbench:

Denne tekst formidler nogle indledende erfaringer, jeg har gjort mig med forskellige notebooks fra GLAM workbenches, der kan bruges til at lave webhistorie. Først skal det siges, at de alle er Jupyter-notebooks, det vil sige at koden er skrevet og kørt i Python. Dog er det ret nemt at sætte Jupyter op og følge med i hvad der sker, da det er veldokumenteret. Tænk på Jupyter-notebooks som Pythons svar på R-Markdown filer, det vil sige dokumentation af hvad koden gør, men skrevet i et helt almindeligt sprog. I det følgende vil jeg gennemgå de notebooks, som jeg har undersøgt. Derudover vil jeg forsøge at komme med eksempler på, hvad de muligvis kan bruges til. Overskrifterne er navnene på Jupyter-notebook filerne. GLAM er en forkortelse for galleries, libraries, archives, and museums. Denne GLAM workbench er lavet af Tim Sherratt og kan findes på dette link: https://glam-workbench.github.io/web-archives/.

Inden vi starter:
Disse workbooks kan alle køres igennem Binder, som er et website, der kan køre Jupyter notebooks uden at man behøver at installere hverken Python eller Jupyter på sin egen computer. Binder kan være et godt sted at starte, men jeg vil anbefale, at man, hvis altså man finder materialet brugbart sætter sig ind i at starte en Jupyter notebook fra sin egen computer, da det giver en noget hurtigere opstart end Binder.

Helt overordnet virker de fleste af disse notebooks til at behandle enkelte webpages. Det som de undersøger er fx udviklingen på en enkelt hjemmeside.

GLAM workbench tager udgangspunkt i Memento API og CDX API (Kort sagt, to forskellige måder at tilgå arkiveret web. Man kan sige, at API'erne er to forskellige sprog, man kan kontakte arkiverne igennem).

GLAM workbenchs notebooks virker med garanti sammen med følgende webarkiver:

- Internet Archive (IA)
- UK Web Archive
- New Zealand Web Archive
- Australian Web Archive

GLAM workbench ligger vægt på, at webarkiver ligner hinanden, så de vil muligvis kunne bruges på andre arkiver også. I mine tests af nogle af notebooksne har jeg gjort brug af IA og har tit søgt efter disse tre sites:

- wellcomecollection.org
- wellcomelibrary.org
- wellcomeimages.org

De følgende overskrifter kan se noget mærkelige ud, men frygt ej. Overskriften er til hjælp. Den første del af overskriften er workbookens overskrift, mens den anden del er workbookens fil-navn.

Getting data from web archives using Memento / memento.ipynb:

Denne notebook introducerer til MementoAPI'en, som de fleste af disse GLAM notebooks er bygget op omkring. Memento API kan grundlæggende finde tre ting til os: Timegates, Timemaps og Mementos. Her vil jeg kort skitsere hvad de forskellige kan:

Timegates kan bruges til at finde den arkiverede version af hjemmesiden, der ligger tættest på det tidspunkt, som man vil undersøge. Hvis man fx er interesseret i hvordan en bestemt hjemmeside så ud den 1. januar 2006, vil timegaten finde den arkiverede version, der ligger tættest på.

Timemaps kan bruges til at danne et overblik over hvornår enkelte URLs, er blevet arkiveret i et bestemt arkiv. Den kan give resultaterne som links, json eller cdxj. Det de enkelte arkiver, der har bestemt hvordan resultaterne kan hentes.

Mementos kan man bruge til at ændre, hvordan resultaterne bliver præsenteret. Fx kan man se det originale website ved at lave små ændringer.

Som nævnt tidligere, er det memento, der ligger bag nogle af GLAMnotebooks. Disse filer er altså bygget op omkring mementos funktioner og forsøger at bruge dem til en masse forskellige analyser i de kommende notebooks, men først noget om den anden API, der bruges.

Exploring the Internet Archive's CDX API / exploring_cdx_api.ipynb:

Denne notebook behandler en anden API end Memento. CDX bruges med andre GLAM notebooks og bruges altså på en lidt anderledes måde end Memento. I både denne notebook og i memento.ipynb er der links til hvilke af de andre notebooks, der er bygget ud fra den respektive API.

I denne notebook er der en introduktion til, hvordan man kan få alle sider på et bestemt domæne med i sit dataset. Generelt handler denne notebook om, hvordan man kan gøre brug af CDX API. Som jeg tænker det, er funktionerne i CDX bedre anvendelige på et stort datasæt, hvorimod Memento er bedre at bruge, når man vil gå i dybden med et par sites. CDX kan nemmere bruges til at finde udviklingstræk, end Memento kan. Med dens *minitypes-funktion* kan man filtrere hvilke data fra hjemmesiderne, man er interesseret i, hvilke filer der skal tælles med og hvilke der ikke skal.

Comparing CDX APIs /comparing_cdx_apis.ipynb

- Sammenligner forskellige webarkivers CDX API'er

Denne notebook viser hvilke små forskelle, der kan være i data hentet fra forskellige API'er. Fx har noget af dataet forskellige titler. Det er demonstration af de begrænsninger, der findes ved de forskellige API'er. Man kan se denne notebook som en guide der kan bruges, hvis man er i tvivl om, hvilken API der bruges i webarkiverne, og hvordan disse API'er adskiller sig fra hinanden.

Getting all available snapshots of a particular page from the Internet Archive – Timemap or CDX? / getting_all_snapshots_timemap_vs_cdx.ipynb

- Denne notebook henter data gennem CDX og Memento og undersøger om man får den samme data frem, der sammenlignes fra Internet Archive. Det er altså stadig en sammenligning af API'erne.
- Notebooken konkluderer, at man får den samme data ud fra begge API'er. Ens valg af API handler om, hvordan man foretrækker at arbejde, og hvad der er hurtigst. Umiddelbart ser CDX ud til at være næsten dobbelt så hurtig som timemap var. Dette kan have betydning for større datasæt.

Get the archived version of a page closest to a particular date / get_a_memento.ipynb

- Simpel notebook, der hurtigt finder linket til den arkiverede udgave af en side, der ligger tættest på i tid, til det tidspunkt, man selv har specificeret.
  Denne notebook kan bruges til at se, hvordan et website har set ud på et givent tidspunkt. Hvis ikke det tidspunkt man søger efter er i arkivet, finder den det tidspunkt, der er tættest på.

Find all the archived versions of a web page / find_all_captures.ipynb

- Indeholder både en Memento og en CDX metode til at finde alle arkiverede versioner af et website. Hvis man vil bruge dataen til mere, skal man så herfra lave det om til en dataframe og så arbejde videre med dele af de næste workbooks, alt efter hvad man gerne vil med dataen. Man kan sige at denne notebook er en "import" notebook. Den bruges til at finde den data frem, man har tænkt sig at arbejde videre med.

Harvesting collections of text from archived web pages / getting_text_from_web_pages.ipynb

- Denne notebook kan udtrække tekst fra alle forskellige arkiveringer af websites, som man så kan arbejde videre med. Dette kunne f.eks være en indgang til at se, hvordan forskellige websites har opfordret til interaktion mellem bruger og site, ved at bede den om at hente sites, hvor ord vi forbinder med interaktion fremgår.
- Denne notebook indeholder både en Memento-metode og en CDX-metode.

Når man bruger notebooken, skal man være opmærksom på, hvad det er man sætter computeren til at hente ned, hvis man beder den om at hente alle hits fra dr.dk, kan det godt fylde meget. Derudover er det vigtigt, at man installerer alle de pakker, der skal bruges, før man begynder at indlæse notebooken (Der er ret mange pakker, som er nødvendige her). Dette er nødvendigt hvis man arbejder lokalt og ikke i Binder.

Det er kort fortalt en notebook, der kan hente tekst fra flere websites, som man så kan arbejde videre med, endnu en "import" notebook.

Harvesting data about a domain using the IA CDX API /n harvesting_domain_data.ipynb:

Denne notebook indeholder kode, der kan høste alt data om et domæne og dets subdomæner. Den giver data, som minder om metadata, det er altså data om sitet og ikke på sitet.

Find and explore Powerpoint presentations from a specific domain / explore_presentations.ipynb

Her forsøger notebooken at finde alle PowerPoints fra en arkiveret hjemmeside. Introducerer "filter" i CDX querys.

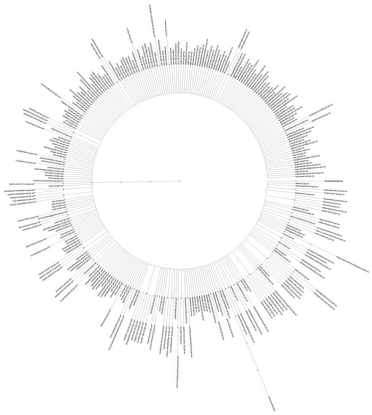
Notebooken indeholder en del elementer. Her er notebookens indholdsfortegnelsen:

1. Harvest capture data
2. Remove duplicates from capture data and download files
3. Convert Powerpoint files to PDFs
4. Extract screenshots and text from the PDFs
5. Save metadata, screenshots, and text into an SQLite database for exploration
6. Open the SQLite db in Datasette for exploration

Denne notebook burde også kunne indstilles til at finde andet end PP. Dette kan gøres ved at lave lidt om i det regulære udtryk, som definere søgningen, her søger man efter dokumenternes endelse, den ville man kunne ændre til .pdf, .jpg eller noget helt tredje. Når man vil søge disse filer frem på et helt domæne og forsøge at downloade dem, skal man forvente, at det tager tid. Der er en del der handler om at konvertere filerne til PDF, som er smart, men lidt krævende. Generelt kan man sige, at denne notebook både indeholder importerende elementer, men også analyserende elementer.

Exploring subdomains in the whole of gov.au / harvesting_gov_au_domains.ipynb:

Denne notebook er den i hele GLAMnotebooks der arbejder på størst materiale. Det vil tage et par timer at høste alt dataen fra IA. Derudover introducerer notebooken til hvordan man kan lave dendrograms, som er en type af visualiseringer, som ses meget småt her til højre. Den version, som notebooken laver, kan man zoome ind på.



Notebooken henter alle subdomæner på hele sitet gov.au, det vil sige, at den henter alt data fra hele det offentlige australske system. Dette er grunden bag notebookens langsommelige process.

Compare two versions of an archived web page / show_diffs.ipynb

Notebook, der kan sammenligne et website over tid. Den sammenligner på følgende parametre: Metadata, statistik, links, ændringer i tekst og screenshots. Notebooken sammenligner altså sitet på mange forskellige parametre. Det eneste man skal gøre er, at finde de to tidspunkter man vil sammenligne i IA.

Using screenshots to visualise change in a page over time / screenshots_over_time_using_timemaps.ipynb:

Jeg kan ikke få denne notebook til at køre optimalt på min computer. Det er ærgerligt, da den skulle give et godt overblik over, hvordan en arkiveret hjemmeside har ændret sig over tid. Den fremstiller en fin visualisering af alt den data, man har kunnet finde i show_diffs.ipynb notebooken.
Tanken bag notebooken, er at den sætter forskellige screenshots af den samme hjemmeside op ved siden af hinanden, så man nemt kan se udviklingen på sitets frontend.

Display changes in the text of an archived web page over time / display_text_changes_from_timemap.ipynb:

Nem notebook, der giver alle tekst ændringer på et bestemt site over tid. Man kan hurtigt skabe sig et overbliv over, hvad der er tilføjet og slettet hvornår. Som notebooken er lavet, sammenligner den alle captures af et site. Muligvis man kan få den til at vise udviklingen på årsbasis, hvis man ikke vil have ændringer fra dag til dag.

Find when a piece of text appears in an archived web page / Find_text_in_page_from_timemap.ipynb:

Denne notebook skal køres i app mode. Når man kører den i appmode, skal man egentligt bare indsætte sine data i boksene og så vente på at "appen" gør sit arbejde. Med notebooken kan man søge på bestemte ord og se hvornår disse ord har været til stede på et site. Man kan fx se hvornår ord er tilføjet, men man kan også se hvis

ens søgeord fjernes igen. Kunne måske være et supplement til en sproglig analyse af et site.

Observing change in a web page over time / change_in_a_page_over_time.ipynb:

Denne notebook kan vise ændringer over tid. Notebooken undersøger mere selve URL'ens ændringer og ikke indholdet. Fx undersøger den forholdet imellem fremkomsten af http- og https-links (Hvor den generelle tendens er at man på et tidspunkt går over til https). Derudover undersøges om sider er besøgt med eller uden www. før domænenavnet. http og www status kobles sammen med de HTTP Status koder, som er registreret i arkivet. Man kan altså se om siden har været tilgængelig, om den har linket videre, været midlertidigt nede osv. Mine iagttagelser var, at siderne wellcomecolletion.org og wellcomelibrary.org havde gennemgået den samme ændring. Deres http sider begyndte at gå fra at give en 200 kode (*Siden tilgås*) til at give en 301 kode (*Siden er permanent flyttet*). Dette skete mens man på https siden samtidigt fik en kode 200, altså at siden findes. Her tyder det på, at siderne er blevet mere sikre, da de er overgået til https fra http.

Med wellcomeimages.org var sagen lidt anderledes. Her så jeg først den samme udvikling med overgang til https, men så begyndte begge sider at linke videre til en anden side. Dette skyldes, at wellcomeimages.org slet ikke eksistere mere på den URL. Istedet kan den findes på https://wellcomecollection.org/works? wellcomeImagesUrl=/. Med denne notebook kan man altså undersøge følgende:

- Udvikling i sikkerhed —> Overgang fra http til https
- Er websites flyttet —> Hvis både http og https versionen sender en videre kunne dette være en mulighed

Alternativ til GLAM workbench: Archives Unleashed

Archives Unleashed projektet, som findes her https://archivesunleashed.org/, kan være svært at gå til i første omgang. Jeg vil anbefale at man starter med deres notebooks. Disse er dog ret komplicerede og kræver kendskab til Python pakken PySpark, før det giver mening at forsøge at køre dem lokalt.
Heldigvis er disse notebooks lavet, så de kan køres i Google Colab. Det betyder, at man ikke behøves at have Python installeret og kan køre dem direkte i sin browser, dog skal man her væbne sig med tålmodighed, da det ikke altid er lige stabilt.

Helt overordnet kræver Archives Unleashed et temmeligt godt kendskab til Python og Jupyter notebooks. Dette kan man kompensere lidt for ved at bruge GoogleColab, men der er ikke så meget tekst der hjælper en på vej her. Ydermere tilbyder Archives

unleashed indgange på forskellige niveauer. De tilbyder fire forskellige tilgange, som kan ses i skemaet nedenfor:

Tilgange til materialet hos Archives Unleashed

Name	Niveau	Info	Hvem er det ideelt for?
Archives Unleashed Cloud	Begynder	Skyen er en webbaseret platform, der gør det muligt at arbejde med Archive-it materiale. Kræver at man kender en bibliotekar med adgang til Archive-it.	Bibliotekarer Forskere
Archives Unleashed Notebooks	Begynder/øvet	Jupyter notebooks. Tilbyder mange forskellige visualiseringer af data, som kan tilgås igennem notebooks.	Forskere
Warclight	Øvet	Warclight er en søgemaskine man kan bruge til at gå på opdagelse i webarkiver. Den kan bruges til at finde webarkiver. Den er nem at bruge, men svær at sætte op på sit eget materiale.	Arkivarer Bibliotekarer
Archives Unleashed Toolkit	Øvet	Toolkit er back-end versionen af cloud. Den er ideel til endnu bredere databehandling, end hvad cloud'en kan tilbyde.	Arkivarer Bibliotekarer Forskere

Jupyter Notebooks hos Archives Unleashed:

Her vil jeg forsøge at skabe et overblik over hvilke elementer, der indgår i Archives Unleasheds (ArcUnl) notebooks. Som nævnt ovenfor anbefaler jeg, at man kører disse notebooks i GoogleColab, da man sådant er fri for at installere en masse Python pakker lokalt. GoogleColab og Binder, der blev brugt tidligere kan nogenlunde det samme. Colab er googles svar på et miljø, hvor flere kan arbejde på den samme kode.

parquet_pandas_example.ipynb

I denne notebook er der en del eksempler på hvordan man kan hive data ud af websites. Fx kigger notebooken på hvilke billeder der har været brugt mest på sitet og kan trække billedet ud, så man rent faktisk kan se det billede. Det undersøges også, hvilke medietyper der har været hyppigst brugt i arkivet. Derudover undersøger den hvilke domæner der er mest brugt i arkivet. Det kan også undersøge hvornår siderne er blevet crawlet.

Udover denne notebook er der to PySpark notebooks, som er en anden måde at arbejde med Python, for at få dem til at gøre lokalt, skal man være tålmodig og villig til at sætte sig ind i PySpark pakken til Python.

Andre ideer:

Hvis man vil lave analyser af tekst fra arkiveret web, kunne det være en mulighed at bruge N-grams, der undersøger hvilke ord der står før og efter det ord men søger efter. Dette kunne give en bedre forståelse for den kontekst ord optræder i.