Networks of Net Literature: Modelling, Extracting and Visualizing Reference-Based Networks in the DLA net literature corpus

Mona Ulrich, Claus-Michael Schlesinger, Pascal Hein, André Blessing
Science Data Center for Literature (German Literature Archive, University of Stuttgart)

ELO Conference 2021, Aarhus University and Bergen University

Abstract
In this essay, following (1) the introduction, we will (2) show explorative examples of network graphs extracted from net literature works that are part of a larger corpus of archived net literature and elaborate the question how these can be useful for further research, (3) describe the corpus we are working with for developing our approach, (4) explain our model of references in web sites, (5) describe the software module Warc2graph which extracts references and thus network information from the archived WARC files in the corpus.

1 Introduction: Hyperfiction itineraries and network topology

“Zeit für die Bombe” (Berkenheger 1997, “Time for the Bomb”) is a complex hyperfiction by Susanne Berkenheger, where the reader follows Veronika and three other protagonists through the narrative evolving around a bomb in a suitcase. The work is built of a large number of interlinked snippets. Snippets can lead to just one more snippet or they can lead to more than one. The reader thus constructs the line of events by choosing where to go next when presented with more than one option. Text snippets can have multiple incoming links and multiple outgoing links. Reading itineraries can lead to certain snippets more than once. Thus, each reading constructs a specific narrative through specific choices at these hypertextual intersections. Any individual itinerary will likely lead to some intersection snippets more than once, which gives the reader the possibility to explore the other paths ignored before. At the same time, the number of snippets is large enough that the overall structure of the text is rather difficult to deduce from following more and different itineraries. The text adresses the reader explicitly in several reflexive passages where a choice leads to a snippet that starts with a comment of that choice or musings about the motivation of the reader. Passing by a specific intersection several times during one itinerary through the text also reflects on the narrative structure and the constructive function of the reader and their reading. It points out the contingency of a certain choice. And with each different route a reader explores, they sketch a trace into their memory, which might lead to a map of the story, should that sketch be written down.

The textual and narrative structure of "Zeit für die Bombe" can be approached both from the perspective of the itinerary experience and from the perspective of the overall structure that can be deduced by mapping all options offered by the text. Modeling networks of net literature and interpreting these networks of net works needs to take into account both perspectives and not confuse the network topology data and visualization with a stable version of an otherwise dynamic object.[1] At the same time, the network structure of a literary text can provide research approaches with specific insights and help navigate the text.

The surge of web archives and the need for analytics approaches in recent years have led to the development of frameworks that provide functions for the extraction of reference information from WARC files, notably the Archives Unleashed Toolkit (Lin et al. 2017) and LinkGate (Eldakar and Alsabbagh 2020). Both frameworks extract reference information from the WARC header. References that were not written into the WARC header at crawl time are lost. Both frameworks provide functions for large scale analytics, where efficiency is a major factor. For our research on net literature, we need detailed representations of reference networks. In this scenario, best outcome, meaning the software catching all references that exist in a WARC file, is more important than computing efficiency. Our software module Warc2graph[2] provides multiple methods for analyzing WARC header and payload data that can be combined in order to achieve most detailed results. Since this is computationally costly, Warc2graph is suitable for analyzing single site objects and small to medium web archives (as opposed to analyzing very large web archives).

[1] We apply the notions of map and route as described by Michel de Certeau (de Certeau 1988) in relation to architectural and geographical spaces. See Landow (1997) for hypertext theory and Ciccoricco (2004) for a more detailed view on the discussion about cognitive maps and topology of hypertext narrative.

[2] Code for all releases will be published to https://github.com/dla-marbach/warc2graph. The package can be installed from the Python Package Index, you can find it under https://pypi.org/project/warc2graph/. A current snapshot (Version 0.1.1) in regard of the publication of this paper can be obtained via Zenodo, DOI: 10.5281/zenodo.4742254 (Hein et al. 2021).

2 Analysis support with Warc2graph

Warc2graph was developed for analysing literary objects published on the Web containing references in HTML elements. The tool extracts information about the resources and references within a literary object and builds a network graph based on that information. With resources, we mean reference targets like images or a HTML page. We consider a network graph to be useful for many (not all) analytical approaches, because the web as a medium offers so many possibilities to define and implement a literary work‘s structure.

2.1 Examples

The mainstream web is influenced by usability principles to better distribute information and products. Good usability should prevent users from leaving when they get lost due to bad design. (Nielson 2012) But the creative use of the medium does not always adhere to usability principles. Here it can happen that websites are even intended to evoke a feeling of being lost in the user, for example, when authors work with hidden links, wild redirects or confusing link structures. As a result, it can be difficult for researchers to get an overview of the site and the resources it contains.

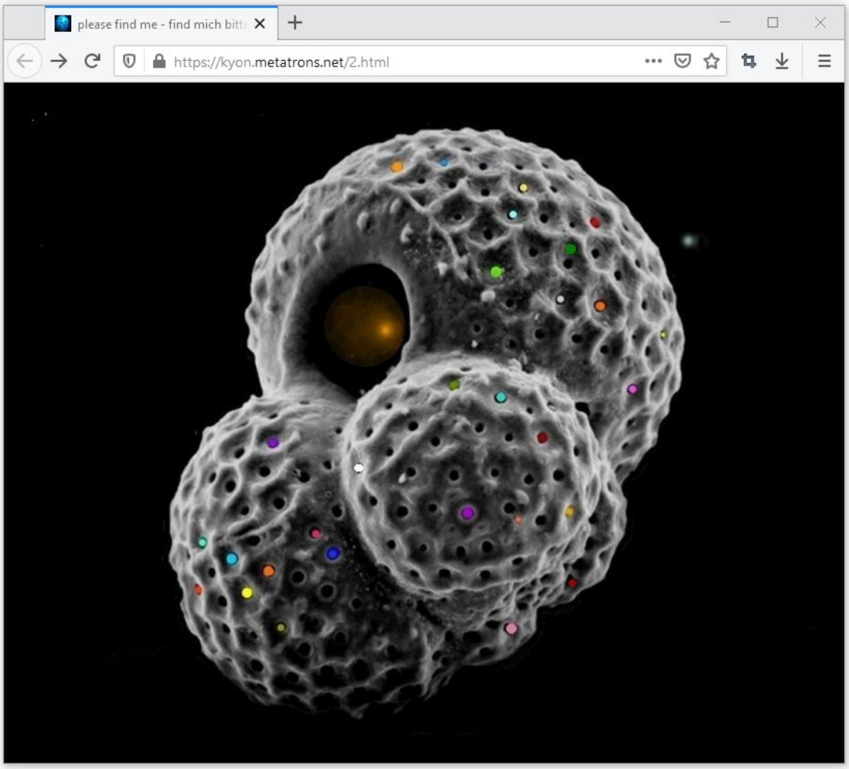

please find me - find mich bitt
https://kyon.metatrons.net/2.html

Figure 1: Kyon‘s Metapage, Kyon, 1998, live web, screenshot
https://kyon.metatrons.net/2.html, „please find me - finde mich bitte, ich warte schon so lange“

The page “please find me - finde mich bitte, ich warte schon so lange” (Kyon 1988) is the centerpoint of the work “Kyon‘s Metapage” by the author Kyon. It contains 31 references to resources, 29 of which are HTML documents. The references use the `<area>` HTML element that defines a clickable area on the page by specifying coordinates, shape and size. By setting a target address using the href attribute, the `<area>` tag becomes a hyperlink. (W3C: 4.8.12) With this method, invisible links can be set everywhere. In Kyon‘s Metapage, the links are placed above the colored elements in the image. Hidden links can be a challenge for researching a particular work. They can only be accounted for methodically by reading the source code, either manually or automatically. Information about the resources and references within the work helps researchers to make sure all references have been discovered without needing to analyze the source code manually page by page.

Another example is the work “btong” from Michael Kaiser. (Kaiser 2001) It contains 211 interlinked HTML documents. The network structure is shown in Figure 2. The work has clear boundaries and contains only work- and domain-internal references and resources. The HTML elements `<a>`, `<frame>`, and `<img>` were used to reference resources. All image resources (nodes) and image references (edges) have been removed from the graph data, since the focus here is primarily on the HTML documents. As as result, all nodes in the graph represent HTML documents and all edges represent references through `<a>` and `<frame>` elements. In this graph the size of a node is determined by the number of its incoming references (larger node size means more incoming references). The visualization was made Gephi using the Yifan Hu‘s Proportional algorithm. (Hu 2005)

The visualization reveals that there are central and peripheral resources. It can be seen that there are documents with either incoming and outgoing references and documents with only one incoming reference, which are located at the periphery of the graph. Only a few documents have very many references and are shown as hubs located more to the center of the network.

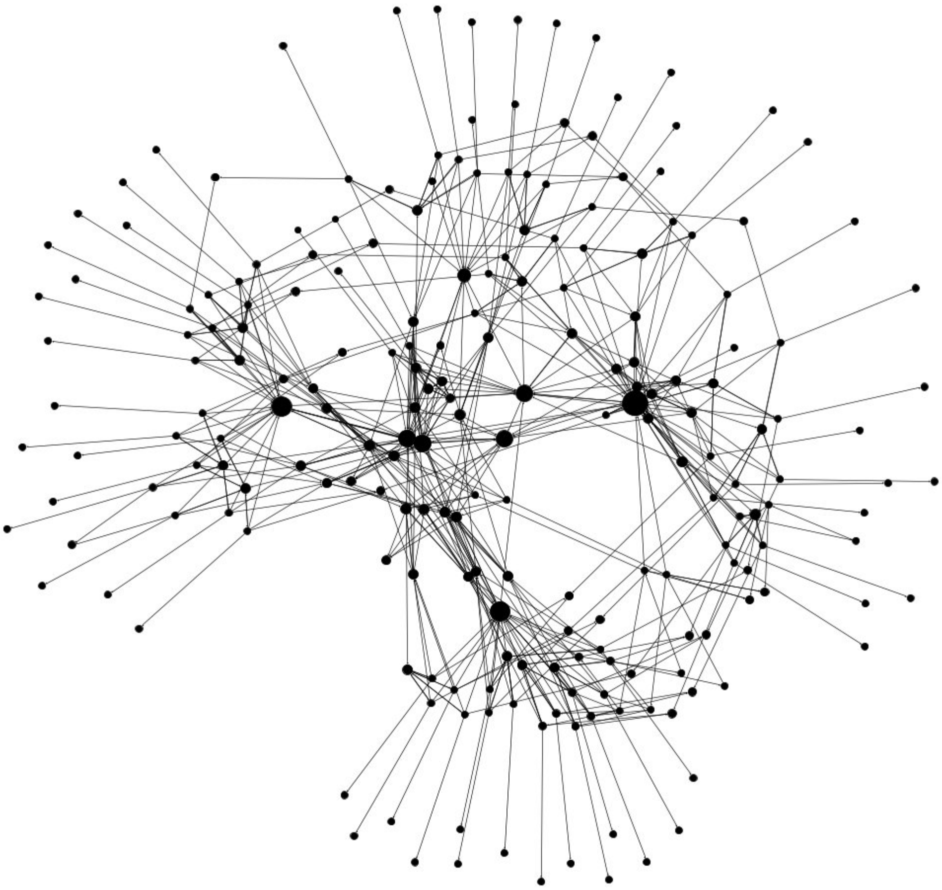


Figure 2: btong, Michael Kaiser, 2001, visualization with Gephi, http://www.polyaethylen.de/btong/btong.htm

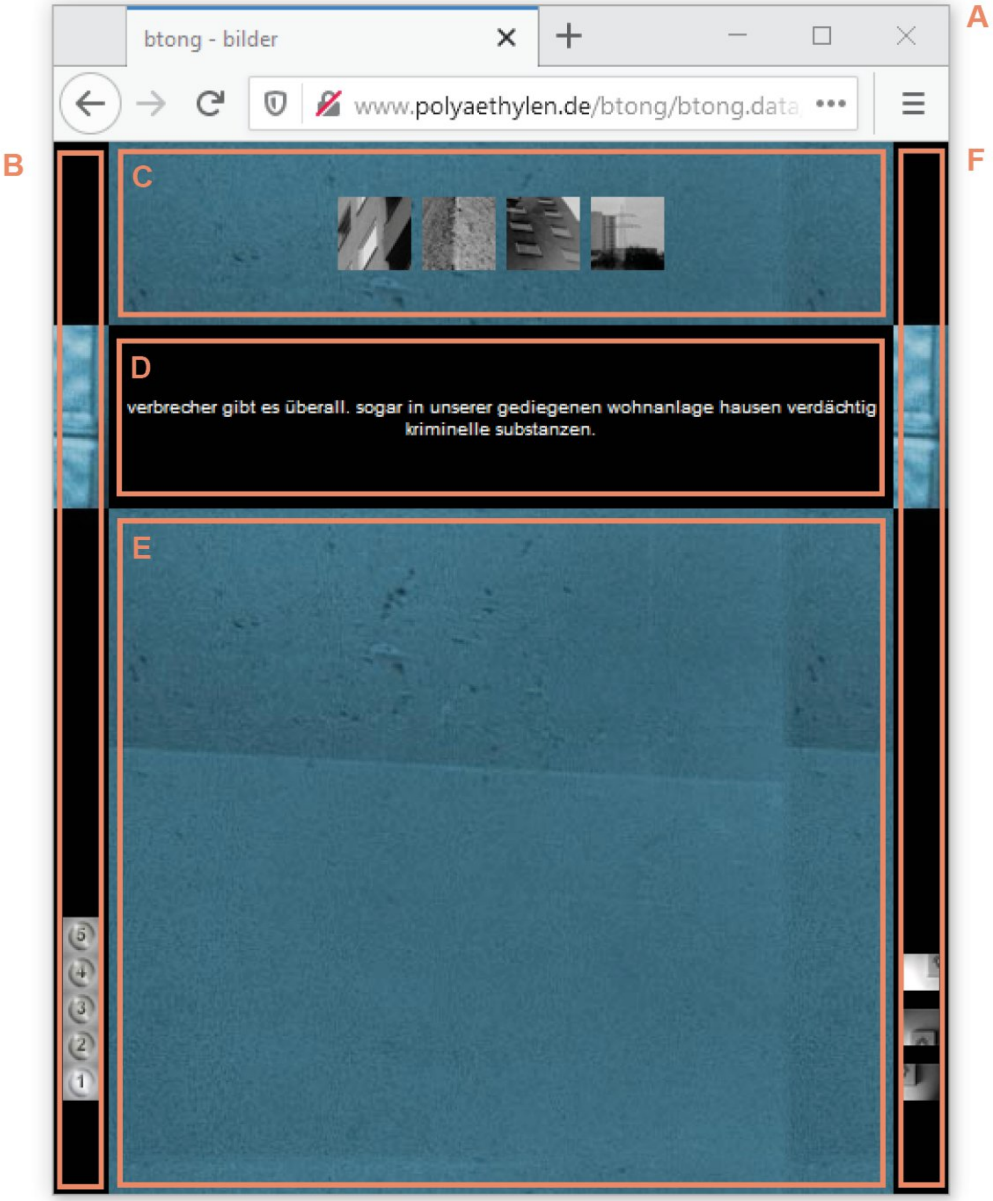
A network visualization can open up alternative analytical approaches, e.g. to pinpoint the resources that are referenced most often and analyze if and why they play a central role. However, such an analysis cannot be adequately conducted without the rendered work on which it is based. Additionally, from our experience it is useful to initially record which HTML reference elements exist in the work and whether it is contains `<frame>`, `<iframes>` or other elements which allow nested browsing contexts.

The edges in our graph data contain information about the kind of reference. Some visualization tools can read and interpret these attributes (others, like Gephi, can‘t). A browsing context is usually a single browser window or a browser tab. Within a browsing context, if a link is clicked, the web page will change. However, HTML elements like `<frame>` and `<iframe>` can initiate additional browsing contexts - so called nested browsing contexts. (W3C: HTML 5.2:6) When working with a network visualization, it is helpful to be able to see in the graph how many browsing contexts the work contains or spawns and which resources will be displayed in which browsing context. The problem of nested browsing contexts is rather complex. With the current version of our software module, browsing contexts are not automatically represented by the graph, but can be identified through the reference type represented by the edge attributes.[3] If a work contained only references via the `<a>` element and the target attribute contains the default value "_self", all pages stay in the same browsing context. The sequence structure of the pages could be read from the graph and it would be easy to get an idea about the rendered work. The more (nested) browser contexts a work consists of, the more difficult it is to read its graph similar to the work.

The work "btong" includes 390 references between HTML documents via the `<frame>` element and 206 via the `<a>` element. As mentioned above, the work contains 211 HTML documents, which means that some documents belong to more than one browsing context. The network visualization is therefore complicated to read. The work interpreted by the browser now helps to explain why some resources are referenced much more often than the rest. The work takes place mainly on pages like the one shown in Figure 3. These pages can be characterized by the fact that they are always referenced through `<a>` and contain five references using the `<frame>` element, spawning multiple browsing contexts. In Figure 3, the different browsing contexts are marked and numbered. Table 1 lists the HTML documents belonging to Figure 3 and their in- and outdegrees. The HTML document contained in the browsing context on the right (ID F) is referenced 20 times, making it the second most referenced resource within the entire work. The high number of incoming references of this resource can be explained by the frames structure - many web pages include the same HTML document for navigation. This means that the resources for navigation are more interconnected and appear more important in the graph. The HTML documents containing literary texts are also referenced via `<frame>`, but usually only once and are therefore mostly located in the periphery of the network.

In this example, the network visualization primarily helped to understand the technical structure. It supported further content and structural analysis, but only after the network graph in relation to the technical structure of the work had been examined and understood.

[3] In order to take the browsing contexts into account for out of the box visualization with warc2graph, further implementations would have to be made: In addition to the information about specific HTML elements analyzed by warc2graph, the target attribute must be evaluated. The target attribute determines in which browsing context the referenced resource should be opened. Possible values are "_self", "_parent", "_top", "_blank" and "<XML name>". The value "_blank" opens a new browser window or a new browser tab (depending on the browser settings). Also, visualizing nested browsing contexts in a single network visualization is a complex problem that would need to be solved. See chapter 6 Next Steps regarding some ideas for further development of a graph-based approach to WARC analytics.



verbrecher gibt es überall. sogar in unserer gediegenen wohnanlage hausen verdächtig kriminelle substanzen.

Figure 3: btong, Michael Kaiser, 2001, live web, screenshot, edited: marked browsing contexts http://www.polyaethylen.de/btong/btong.data/bbtong/bbtong.htm

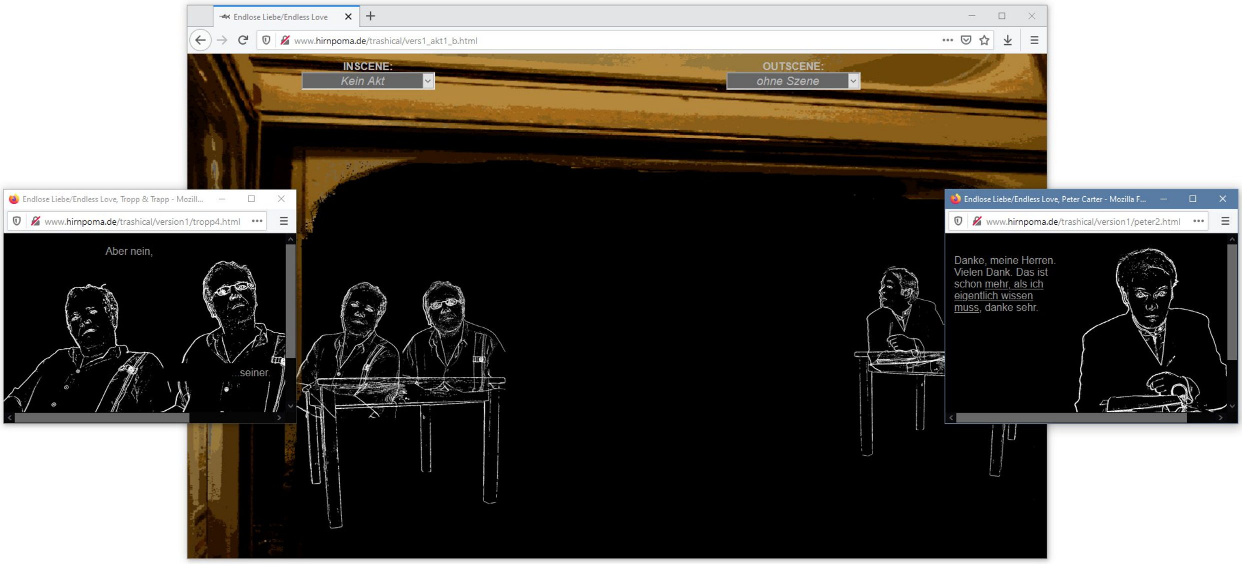
ID	Resource	Indegree	Outdegree
A	http://www.polyaethylen.de/btong/btong.data/bbtong/bbtong.htm	16 <a>	5 <frame>
B	http://www.polyaethylen.de/btong/btong.data/bbtong/pali01.htm	5 <frame>	5 <a>
C	http://www.polyaethylen.de/btong/btong.data/bbtong/pmo01.htm	5 <frame>	4 <a>
D	http://www.polyaethylen.de/btong/btong.data/btong/amm00.htm	15 <frame>	0
E	http://www.polyaethylen.de/btong/btong.data/btong/a0000000.htm	4 <frame>	0
F	http://www.polyaethylen.de/btong/btong.data/btong/are01.htm	20 <frame>	3 <a>

Table 1: btong, Michael Kaiser, 2001, URLs, in- and outdegrees

Frank Klötgen‘s and Joachim Schäfers‘s online musical “Endlose Liebe. Endless love” (Klötgen and Schäfer 2005) also consists of several browsing contexts (see Figure 4). Pop-up windows represent the actors in the play. And as actors would do, the pop-up windows talk to each other. By clicking a link within one pop-up window, the content in another pop-up window changes. The network graph would be complicated to understand, but this is not the main issue. Most apply a mix of the <a> element and JavaScript functions.

```
<a href="javascript:window.open('peter1.html','gegentext', 'width=449,height=291');
textausblenden('1_t3'); textausblenden('1_tr2');" onFocus="if(this.blur)this.blur()">jetzt
noch</a>
```

The JavaScript method `window.open` is part of the <a> tag and opens a HTML Document in a new window. This reference, or rather the referenced resource, will not be handled correctly by Warc2graph. Warc2graph would create a node with the value in href. The string would not be parsed to look for the HTML document within the JavaScript method `window.open`. The resulting network graph is incomplete.



Endlose Liebe/Endless Love
www.hirnpoma.de/trashical/vers1_akt1_b.html
INSCENE:
Kein Akt
OUTSCENE:
ohne Szene
Endlose Liebe/Endless Love, Tropp & Trapp - Mozill...
www.hirnpoma.de/trashical/version1/tropp4.html
Endlose Liebe/Endless Love, Peter Carter - Mozilla F...
www.hirnpoma.de/trashical/version1/peter2.html

Aber nein,

...seiner.

Danke, meine Herren. Vielen Dank. Das ist schon mehr, als ich eigentlich wissen muss, danke sehr.

Figure 4: Endlose Liebe. Endless love, Frank Klötgen, Joachim Schäfer, 2005, live web, screenshot http://www.hirnpoma.de/trashical/

2.2 Summary

Network graph data and visualization assists research by providing an overview of the work (How much resources are included in the work?) and its structure (How are resources interlinked? What incoming and outgoing references does a particular resource have?).

The graph makes it possible to examine the structure of a work. (How to characterize the resources within the network? Where are central parts of the work, and what is their function?). Information about a site‘s structure can prove helpful when analyzing narrative structure, as it can support sorting out hard-coded boundaries and connections between textual elements. For larger sites, network information makes it possible to identifiy key elements and navigate the work. Last not least, Warc2graph literally analyzes a work, in that it transforms WARC data into a graph structure, which makes it possible to navigate, search or sort elements according to the type of reference or the type of node, e.g. images.

Warc2graph does not recognize all possible references. The system has limited outcomes if references are constructed using complex JavaScript, Java or any proprietary binary format like Shockwave or Flash. Before working with Warc2graph (or any WARC analyzer), it is necessary to define expectations taking into account the analytical scope of the system.

3 Corpus

The literary objects on which we developed warc2graph are part of the collection *Literature on the Web* at the German Literature Archive (DLA) in Marbach. The collection contains archived literary net objects with a total of about 500 sources. The objects have been archived from 2008 to 2018.[4] The archived sources are currently accessible on the platform Literatur-im-Netz. (Deutsches Literaturarchiv: Literatur im Netz) Within the collection, works are assigned to the categories literary blogs, literary online magazines and net literature. Classification is based on genre or conceptual characteristics. Technical characteristics of the sources are not decisive for the classification, and yet the sources within a category often share similar technical characteristics. The net literature works in the collection were created between 1995 and 2011. The majority of the works are characterized by common features, which do not necessarily describe the genre of net literature. The websites were often written by the authors themselves and are not based on prefabricated templates. The object boundaries of the works are usually clearly definable and the objects contain several interlinked HTML documents.

The collection mandate of the German Literature Archive Marbach encompasses german literature and contemporary documents of literary and cultural life from 1750 to the present. (Deutsche Schillergesellschaft 2018) Selection criteria for network objects take into account whether an object contains primary texts, is an exemplary representative of individual literary-

[4] Since 2018 webarchiving is temporarily suspended because the technical service provider discontinued the development and operation of the SWBcontent web archiving service.

technical forms and time periods, or has been discussed in the cultural sector. (Deutsches Literaturarchiv: Auswahlkriterien)

The web objects are archived in WARC files. The file stores the resources requested and recieved by the crawler or other web archiving tools, as well as the control communications of common Internet protocols. A WARC file consists mainly of recorded requests and responses, sent by client and webserver, and metadata describing the contained resources. (IIPC: The WARC Format) Each literary net object is stored in its own WARC file. The resources (e.g. HTML, CSS, JavaScript, Java Applets, image files) are in the payloads of the recorded responses from the webserver.

4. Modelling

In order to visualize the structural information as a graph, we needed to define which elements are represented by nodes and by edges. We decided on a graph model where the nodes represent the resources and the edges represent the HTML elements referring from one resource to another.

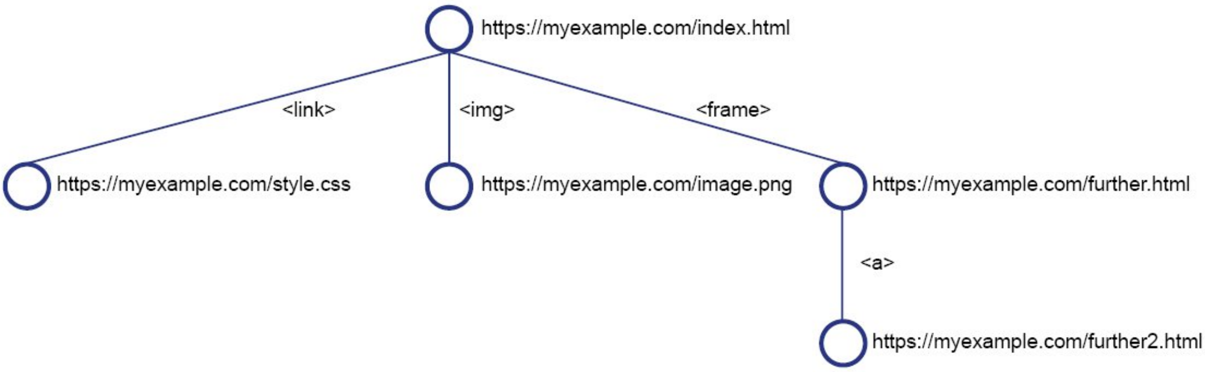

https://myexample.com/index.html
<link>
<img>
<frame>
https://myexample.com/style.css
https://myexample.com/image.png
https://myexample.com/further.html
<a>
https://myexample.com/further2.html

Figure 5: Illustration of the graph model

We colloquially refer to Warc2graph's operation as link extraction, but what we actually do is "reference extraction". Only few HTML elements that we extract are considered links. W3C defines only the following elements as links: `<a>`, `<link>` and `<area>`. (W3C: 4.12) However, they are considered links only if they contain the href attribute. (W3C: HTML 5.2:4.5) There are other HTML elements that also include the href attribute or other attributes that can refer to html files: `<base>`, `<frame>`, `<object>`, `<applet>`, `<embed>`, `<meta>`, `<form>`, `<button>`, `<q>` and `<blockquote>`. So it is not sufficient to search only for HTML elements considered as links.

This bears the question: If we want to collect not only links, how should the selection of elements be extended? It seems clear that the focus is on looking for references to HTML files, but can it be a strict rule to look only for those references? Not for our purpose, because links to image files, PHP files, etc. would be disregarded, e.g. `<a href="nice.img">, <a href="index.php">`. It is also not possible to look only for HTML elements containing absolute and relative HTTP URLs. References to image files or PHP files would be included, but a link to a mail adress (`<a href="mailto:someone@example.org">`)would be excluded. Since we cannot distinguish HTML elements with references to URLs by either the type of referenced object or the reference type, we search for all HTML elements with attributes which can contain references to relative or absolute URLs.

5 Warc2graph functions

Using the python package Warc2graph archived websites stored in the WARC format can be analysed and modelled as network graphs. The package can be obtained from the Python Package Index. It provides a python library that allows a versatile and customizable usage as well as an easy to use command line application.

Our tool reads the WARC file using the library warcio (Webrecorder Project) and accesses metadata and stored resources to find references between resources implemented as HTML tags. To build the network graph all resources – e.g. HTML, CSS, or image files – are modelled as nodes while all references between them are modelled as edges. The nodes are defined using the original absolute URL and the directed edges contain information about the tag that connect the resources.

5.1 Usage

The command line interface can be accessed using the `warc2graph` command. Additionally the path to a WARC file must be passed as a parameter. The tool processes the WARC file and outputs three files: one file containing the data from the network graph stored in the GEXF (GEXF Working Group 2009) format which is based on XML, another file containing multiple visualizations for a first impression of the extracted references and a JSON file containing metadata describing the analysed object as well as the process of creating the network graph (date and time of the creation as well as all parameters used for running the program). Metadata is created automatically and can be supplemented manually. If more than one path to WARC files is passed to the program, extraction results will be merged and presented in one graph. This functionality is provided because WARC files are not supposed to be bigger than 1GB which sometimes leads to websites being archived in more than one file. (IIPC)

Additionally, links to web pages on the live web can be passed, allowing analysis of websites on the live web.

The library `warc2graph` for use in python environments or programs provides the principal function `create_model` that needs a path to a WARC file as input. This function returns a directed graph that is implemented with the open source library NetworkX. (Hagberg et al. 2008) This means that NetworkX can then be used for further computation, e.g. calculate centrality metrics or test for specific properties such as circularity.

5.2 Methodology

Modeling a website as a network graph using Warc2graph is a two-step-process. The function `warc2graph.extract_links` reads the WARC files and extracts all the references while the function `warc2graph.create_network` uses the extracted data to create the directed graph. Using the python library the two steps can be run automatically in the background using the `create_model` function but can also be called separately. In this case the data can be examined and cusomized after each step.

To extract the references from the WARC file our tool iterates over all the entries in this file using the warcio library. All the HTML files are then analysed using three different extraction methods.

1. The most basic approach reads out the metadata stored in the WARC file. Usually all the outlinks found at crawling will be stored here. This method is quite robust and performant but it lacks flexibility. Domains can be filtered, but what was not found in the crawling process can also not be analysed now. To make matters worse storing the outlinks is not defined as a requirement in the WARC specification. Quality and scope thus depends on the tools used for crawling and archiving.
2. In our second approach we analyse the HTML data using the python library BeautifulSoup. (Richardson 2020) All tags contained in the HTML data can be found and checked for references to other resources. References that are generated procedurally by the browser using JavaScript will not be recognized.
3. In order to also evaluate JavaScript the HTML data is processed with the remote controlled headless browser Selenium. (Selenium Project) Opening and controlling this browser as well as evaluating the JavaScript code has a significant negative impact on runtime and is not suited for large archives.

After the first partial step of the extraction of references the data will be stored in a list of tuples containing the URLs of the source and the target of the reference. In the next step the created list is being transposed to a network graph. The data and metadata stored in the graph can be accessed and extended.

The software is built in a modular fashion, so those interested only in the extraction of references can rely on the list of tuples returned by the `extract_links` function. In the same manner, a graph compatible with the graphs created from WARC files can be constructed by passing a list of tuples to the `create_network` function.

6. Next Steps

We have used Warc2graph successfully for an exploration into non-linear text structures and non-linear narrative in specific works of net literature from the DLA corpus. In order to make the approach more robust we hope to expand the scope of our tests to other web archive corpora, do a more in-depth assessment of existing WARC analytics frameworks and better define the specific function Warc2graph can fulfill in web archive analytics workflows. Furthermore, the transformation of WARC files into graph representations opens up archival and research possibilities on a corpus level. We will explore the possibilities of such graph representations for further analytical questions pertaining not only to references, but also to the content of the elements or nodes of each object and of different aggregations up to the level of large web archives. This approach also requires epistemological and aesthetic reflections on graph representations and topologies of either dynamic works of net literature and non-specialized web archives.

References

Berkenheger, Susanne. *Zeit Für Die Bombe*. 1997, http://www.berkenheger.netzliteratur.net/ouargla/wargla/zeit.htm.

Certeau, Michel de. *Kunst Des Handelns*. Merve, 1988.

Ciccoricco, Dave. “Network Vistas: Folding the Cognitive Map.” *Image & Narrative*, no. 8, 2004, http://www.imageandnarrative.be/inarchive/issue08/daveciccoricco.htm#013.

Deutsche Schillergesellschaft e.V. *Satzung*. 2018, https://www.dla-marbach.de/fileadmin/redaktion/Ueber_uns/Satzung_DSG_Stand_2018.pdf.

Deutsches Literaturarchiv Marbach. *Auswahlkriterien Und Verfahren*. https://www.dla-marbach.de/en/library/literature-on-the-web/voraussetzungen/auswahlkriterien-und-verfahren/. Accessed 7 May 2021.

---. *Literatur Im Netz*. http://literatur-im-netz.dla-marbach.de/. Accessed 7 May 2021.

Eldakar, Youssef, and Lana Alsabbagh. *LinkGate: Let’s Build a Scalable Visualization Tool for Web Archive Research*. 23 Apr. 2020, https://netpreserveblog.wordpress.com/2020/04/23/linkgate-update/.

GEXF Working Group. *GEXF File Format*. 2009, https://gephi.org/gexf/format/.

Hagberg, Aric A., et al. “Exploring Network Structure, Dynamics, and Function Using NetworkX.” *Proceedings of the 7th Python in Science Conference*, edited by Gaël Varoquaux et al., 2008, pp. 11–15.
Hein, Pascal, et al. *Warc2graph*. Zenodo, 2021. *Zenodo*, DOI: 10.5281/zenodo.4742254, https://zenodo.org/record/4742254.

Hu, Yifan. “Efficient and High Quality Force-Directed Graph Drawing.” *The Mathematica Journal*, no. 10, 2005, pp. 37–71.

IIPC. *The WARC Format*. https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/. Accessed 7 May 2021.

Kaiser, Michael. *Btong*. 2001, http://www.polyaethylen.de/btong/btong.htm.

Klötgen, Frank, and Joachim Schäfer. *Endlose Liebe. Endless Love*. 2005, http://www.hirnpoma.de/trashical/.

Kyon. *Kyon’s Metapage*. 1998, https://kyon.metatrons.net/2.html.

Lavoie, Brian, and Henrik Frystyk Nielsen. “Web Characterization Terminology & Definitions Sheet.” *Web Characterization Terminology & Definitions Sheet*, 1999, https://www.w3.org/1999/05/WCA-terms/.

Landow, George P. *Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology*. Baltimore: The Johns Hopkins UP. 1997.

Lin, Jimmy, et al. “Warcbase: Scalable Analytics Infrastructure for Exploring Web Archives.” *J. Comput. Cult. Herit.*, vol. 10, no. 4, July 2017, p. 22:1-22:30. *ACM Digital Library*, doi:10.1145/3097570.

Nielson, Jakob. *Usability 101: Introduction to Usability*. Jan. 2012, https://www.nngroup.com/articles/usability-101-introduction-to-usability/.

Richardson, Leonard. *Beautiful Soup Documentation*. 2020, https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Selenium Project. *The Selenium Browser Automation Project*. https://www.selenium.dev/documentation/en/. Accessed 29 Apr. 2021.

W3C HTML Working Group. *4.8.12 The Map Element*. 2011, https://www.w3.org/TR/2011/WD-html5-20110405/the-map-element.html.

---. *4.12 Links — HTML5*. 2011, https://www.w3.org/TR/2011/WD-html5-20110405/links.html.

---. *HTML 5.2: 4.5. Text-Level Semantics*. Jan. 2021, https://www.w3.org/TR/html52/textlevel-semantics.html.

---. *HTML 5.2: 4.7. Embedded Content*. Jan. 2021, https://www.w3.org/TR/html52/semantics-embedded-content.html.

---. *HTML 5.2: 6. Loading Web Pages*. 2021, https://www.w3.org/TR/html52/browsers.htm.

Webrecorder Project. *WARCIO: WARC (and ARC) Streaming Library*. https://github.com/webrecorder/warcio. Accessed 7 May 2021.