Nicht-lineare Narrative in Netzliteratur: Speicherung und Nachnutzung von Forschungsdaten aus der computergestützten Extraktion von Verweisstrukturen in Hypertexten

Claus-Michael Schlesinger[2], Mona Ulrich[1],Pascal Hein[2], André Blessing[2], Nina Buck[3], Björn Schembera[3], Volodymyr Kushnarenko[3], Andreas Ganzenmüller[3], Lisa Kiss[2], Julia Horvat[2] und Oksana Nedostup[2]

[1]Deutsches Literaturarchiv Marbach
[2]Universität Stuttgart
[3]Höchstleistungsrechenzentrum Universität Stuttgart

Das Forschungs- und Infrastrukturprojekt *Science Data Center for Literature* entwickelt und implementiert ein Repository für born-digital-Bestände am Deutschen Literaturarchiv Marbach (DLA) und ein zugehöriges Portal mit Forschungsumgebung. Wir beschreiben einen exemplarischen Forschungsansatz zur Analyse nicht-linearer Strukturen in Netzliteratur, das Teilkorpus "Literatur im Netz" am Deutschen Literaturarchiv (Entwicklungsgrundlage), das Softwaremodul Warc2graph zur Extraktion von Verweisstrukturen aus den archivierten Objekten, die konzipierte Architektur der Plattform für SDC4Lit als Zielumgebung für das Modul sowie weitere Nachnutzungsmöglichkeiten.

1 Einleitung

Literarische Werke im WWW können von technischen und kulturellen Gegebenheiten des Mediums inspiriert sein. Sie zeichnen sich daher oft durch eine besondere Beziehung zwischen literarischem Text und technischem Medium aus. Neben der Bedeutung grafischer und typografischer Gestaltung zählt dazu insbesondere die Hypertextstruktur und Hypermedialität der Werke[5, 11].
Die Verteilung einer Erzählung auf mehrere miteinander verlinkte Webseiten bedingt eine nicht-lineare Struktur, die oft mit nicht-linearen narrativen Strukturen korrespondiert. Linearität und Nicht-Linearität sind dabei auf den sukzessiven Durchgang durch ein Werk im Zuge der Lektüre oder Interaktion bezogen. Zu unterscheiden sind dabei zwei unterschiedliche Perspektiven auf ein

Publiziert in: Vincent Heuveline, Nina Bisheh (Hg.): E-Science-Tage 2021. Share Your Research Data. Heidelberg: heiBOOKS, 2022.
DOI: https://doi.org/10.11588/heibooks.979.c13727 (CC BY-SA 4.0)

Hypertextobjekt: Erstens eine Perspektive, die die Hypertextstrukturen ohne den durch Ein- und Ausgabegeräte sowie hypermediale und Interaktionsfunktionen vorgegebenen Verlauf betrachten, und zweitens eine Perspektive, die diese Aspekte mit einbezieht und die Struktur als Gesamtmenge aller möglichen Durchgänge durch ein Werk versteht, d.h. alle möglichen Durchgänge durch ein Werk von allen Einstiegs- zu allen Endpunkten berücksichtigt. Wir sehen eine nicht-lineare Struktur als gegeben, sobald eine Seite mehr als einen Verweis auf Folgeseiten enthält. Ein linearer Durchgang durch den Gesamttext ist dann nicht mehr möglich. Für narrative Texte folgt daraus auch ein zumindest in Teilen nicht-linearer Erzählverlauf. Nicht-linerare Textstrukturen können überwiegend lineare Erzählverläufe mit alternativen Strängen, variablen Enden und zyklischen Elementen oder durch komplexe Verlinkungen verschiedene Erzählverläufe ermöglichen.

Für eine narrative Analyse im engeren Sinn eignen sich nur Werke der Netzliteratur, die entsprechende narrative Eigenschaften aufweisen. Ein klassisches Beispiel sind hier Hyperfiction-Texte wie *Zeit für die Bombe* (1997) von Susanne Berkenheger. [1, 20] In *Zeit für die Bombe* sind Protagonist*innen, Ereignisse und also eine Geschichte im narratologischen Sinn[15] unzweifelhaft vorhanden. Andere Werke wie zum Beispiel Johannes Auers *Kill the Poem* (1997) orientieren sich dagegen eher an Formen der Konkreten Poesie und liefern keine erzählte Handlung, sondern einen Handlungsraum für spielerische ästhetische Ko-Produktion und Interaktion. Auf einer abstrakten Strukturebene ist die Verweisstruktur für alle Objekte im Bereich "Literatur im Netz" ein für die Analyse relevantes Merkmal, in dem sich technische Funktionen mit ästhetischen Eigenschaften verbinden.[5] Im Folgenden geht es dabei weniger um gegenstandsbezogene literaturwissenschaftliche Fragestellungen, sondern um die archivarischen, objektanalytischen und infrastrukturellen Kontexte und Voraussetzungen, in denen solche Fragen gestellt und bearbeitet werden können. Der Fokus liegt auf den Bereichen Archiv, Analyse und Infrastruktur, wie sie sich im Forschungs- und Infrastrukturprojekt *Science Data Center for Literature* (SDC4Lit) darstellen.[1]

In Abschnitt 2 beschreiben wir das Korpus "Literatur im Netz" des Deutschen Literaturarchivs, das die Materialgrundlage bildet für unsere Modellierung von Verweisstrukturen in archivierten Netzobjekten. Zweitens stellen wir mit `warc2graph` ein Softwarepaket vor, das dieses Modell implementiert und Verweisstrukturen aus archivierten Objekten extrahiert.[2]

Insbesondere für die Analyse von Einzelobjekten und kleineren Korpora ist ein hoher Grad an Genauigkeit wünschenswert. Aktuell verfügbare Ansätze zur Extraktion von Verweisstrukturen aus archivierten Netzobjekten und -korpora zielen auf die Analyse von großen Datenmengen und legen den Schwerpunkt daher auf höchstmögliche Effizienz. [13, 4] Warc2graph zielt dagegen auf höchstmögliche Ergebnisqualität (mit den entsprechenden Abstrichen bei der Performanz). Drittens skizzieren wir die Architektur der Forschungs-

[1] Siehe hierzu das Extended Abstract zum Poster von SDC4Lit in diesem Band

[2] Warc2graph ist als Python-Modul über den Python Package Index verfügbar, aktuelle Versionen werden außerdem im Github-Repository von Warc2graph bereitgestellt, `https://github.com/dla-marbach/warc2graph`. Das Softwarepaket in der zum Zeitpunkt der Veröffentlichung dieses Beitrags aktuellen Version 0.1.1 siehe auch [9]

umgebung, in der Warc2graph für die Unterstützung wissenschaftlicher Analysen der archivierten Gegenstände als Modul eingesetzt werden soll. In einem kurzen Ausblick nennen wir abschließend weitere Nachnutzungsmöglichkeiten, die sich aus unserer Sicht für das Modul anbieten.

2 Gegenstand Literatur im Netz

Unsere Forschungsfrage zu nicht-linearen narrativen Strukturen bezieht sich auf literarische Objekte, die im Web veröffentlicht sind und Verweise mittels HTML-Tags oder JavaScript-Funktionen beinhalten. Entwickelt wurde die Forschungsfrage an den archivierten Netzliteraturwerken der Sammlung Literatur im Netz des deutschen Literaturarchivs in Marbach (DLA). Zu der Sammlung gehören neben Netzliteraturwerken auch literarische Blogs und Online-Magazine. Die Archivierung der Quellen erfolgte von 2008 bis 2018. Die archivierten Quellen sind derzeit auf der Plattform *Literatur im Netz* zugänglich.[3] Die Netzliteraturwerke sind zwischen 1995 und 2011 entstanden. Der Großteil der Werke zeichnet sich durch gemeinsame Charakteristiken aus, die aber nicht zwangsläufig die Literaturgattung Netzliteratur an sich beschreiben. Dazu gehört, dass die Webseiten von den Autor*innen meist selbst geschrieben wurden und nicht auf vorgefertigten Templates basieren. Die Objektgrenzen der Werke sind meist klar definierbar und die Objekte beinhalten mehrere HTML-Dokumente, die untereinander verlinkt sind. Durch die Wahl des Mediums haben die Autor*innen die Möglichkeit, ihre Werke beliebig zu strukturieren, wodurch die manifestierten Entscheidungen bezüglich der Struktur bedeutungsvoll sind und daher ebenso wie die textuellen Inhalte bei der Analyse berücksichtigt werden müssen.

Auch für die literarischen Blogs und Onlinemagazine ist eine Analyse der Seitenstrukturen aufschlussreich, unabhängig von der Ausgangsfrage zu nicht-linearen narrativen Strukturen. Die Visualisierung der Seitenstruktur bietet Forscher*innen einen neuen Überblick über das Objekt. Denn anders als bei makrophysikalischen Objekten, zum Beispiel bei einem Buch oder einer Kunstinstallation, sind die Objektgrenzen und der Aufbau eines Netzobjekts nicht sichtbar. Auf der Startseite einer Website ist nicht ersichtlich ob sie zwei oder zweihundert Unterseiten enthält und wie die Seiten sich zueinander verhalten. Forscher*innen müssten sich diese Übersicht aufwendig erarbeiten - für manche Objekte eine fast unlösbare Aufgabe.

Die Strukturinformationen können im hier vorliegenden Anwendungsfall von archivierten Websites, die im WARC-Format[4] gespeichert sind, auch automatisch ausgelesen werden. Das WARC-Format wurde vom International Internet Preservation Consortium (IIPC) aufbauend auf das ARC-Format entwickelt. Das ARC-Format entstand 1996 am Internet Archive, um gecrawlte Webressourcen besser verwalten zu können.[5] In einer WARC-Datei

[3] [14]
[4] IIPC
[5] [16]

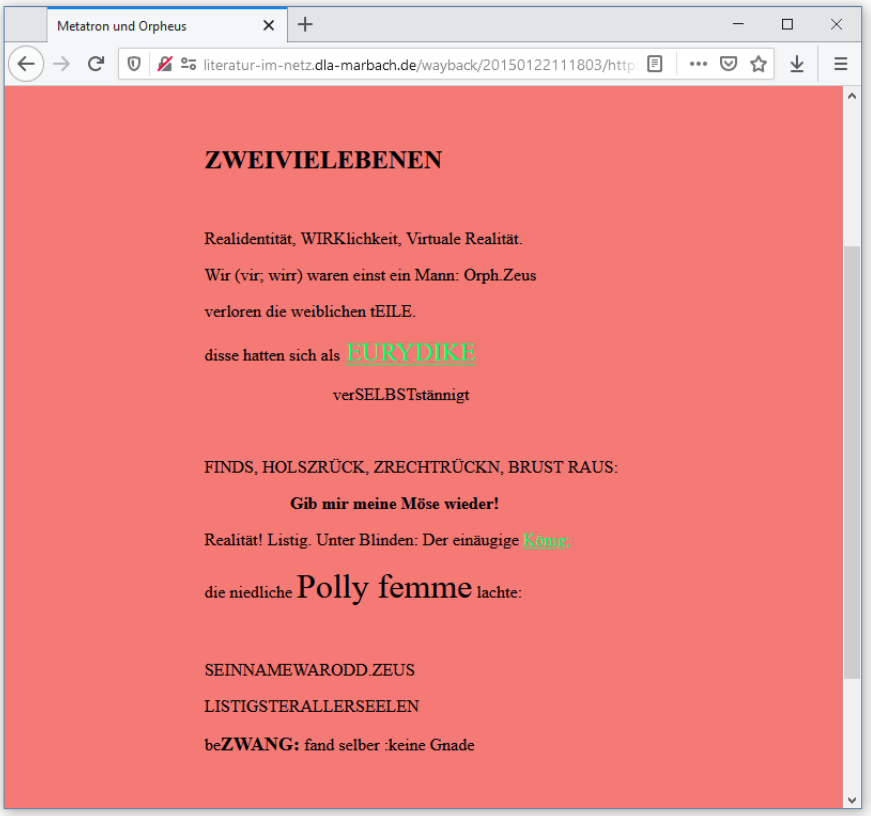

Abbildung 1: Kyon's Metapage, Kyon,1998, `https://metatrons.net/m4.html`, hier Archivversion, Screenshot.

wird die Steuerungskommunikation zwischen Client und Server gespeichert, sowie die vom Client (Webarchivierungstool, Browser) empfangenen Ressourcen.

Die Steuerungskommunikation und WARC-spezifische Metadaten sind als Text und die Ressourcen, sofern sie keine Textdateien sind, in Binärform enthalten. Die Ressourcen werden immer unverändert, das heißt so wie sie empfangen wurden, abgespeichert. Eine WARC-Datei kann mit speziellen Tools, zu nennen sind hier Pywb[6] und OpenWayback[7], wiedergegeben werden. Im Optimalfall wäre der einzige Unterschied zwischen einer originalen Seite im live web und ihrer archivierten Version die URL in der Adresszeile des Browsers.
Die literarischen Objekte am DLA wurden jeweils in eigenen WARC-Dateien gespeichert. Aus einer WARC-Datei können die Strukturinformationen zu einem Objekt mit unterschiedlichen Methoden ausgelesen werden, die teilweise ergänzend eingesetzt werden müssen. Das von uns entwickelte Tool Warc2graph beherrscht mehrere Methoden (einzeln und kombiniert) und ermöglicht dadurch einen Vergleich der Ergebnisse, die durch einzelne oder kombinierte Ansätze gewonnen werden.

[6]Webrecorder Project
[7]IIPC, OpenWayback, `https://github.com/iipc/openwayback`, aufgerufen: 27.04.2021.

3 Warc2graph

Mit Hilfe des Python-Pakets Warc2graph können im WARC-Format archivierte Websites[8] automatisiert ausgelesen und als Netzwerkgraph modelliert werden. Das Paket kann über den Python Package Index bezogen werden. Es stellt sowohl eine in Python importierbare Bibliothek als auch eine einfach zu bedienende Kommandozeilenanwendung bereit. Das Tool öffnet die WARC-Datei mithilfe der Python Bibliothek Warcio[9] und greift auf die Metadaten und die gespeicherten Ressourcen zu, um Verweise zwischen den Ressourcen zu finden, die durch HTML-Tags und deren Attribute bestimmt sind. Für den Netzwerkgraphen werden alle Ressourcen – HTML-, CSS-, Bilddateien und andere Medienformate – als Knoten gespeichert und alle Verweise von und auf die Ressourcen als Kanten. Die Knoten werden über die absolute URL der Ressourcen definiert. Die Kanten sind mit der Information darüber angereichert, welche Art von Verweis sie darstellen. Alle HTML-Tags und ihre Attribute, die URLs enthalten können, werden hier ausgelesen.

3.1 Anwendung

Die Kommandozeilenschnittstelle wird über den Befehl `warc2graph` aufgerufen. Zusätzlich muss der Pfad zu einer WARC-Datei als Parameter mit angegeben werden. Das Tool verarbeitet daraufhin die WARC-Datei und erstellt als Output drei Dateien. Dies ist erstens eine auf XML basierende GEXF-Datei[10], die die Graphdaten enthält. Zweitens werden Visualisierungen erstellt, die einen ersten Überblick zur Struktur der extrahierten Verweise liefern sollen[11]. Dabei werden jeweils drei Netzwerkdiagramme mit drei unterschiedlichen Visualisierungsalgorithmen erstellt, um zu vermeiden, dass ein einziges spezifisches Layout ungerechtfertigte Rückschlüsse auf die Struktur der Website motiviert. Die dritte Outputdatei beinhaltet Metadaten im JSON-Format. Hierbei handelt es sich sowohl um Metadaten, die die ursprüngliche archivierte Website beschreiben, als auch um Metadaten, die den Prozess der Erstellung des Graphen aus der WARC-Datei beschreiben und dabei neben Datum und Uhrzeit auch alle gewählten Parameter beinhalten. Die Metadaten werden automatisiert erstellt, können aber manuell beliebig ergänzt werden.

Alternativ können auch mehrere WARC-Dateien übergeben werden. Hierbei werden die Ergebnisse zusammengefasst und wie ein Archiv einer einzigen Website behandelt. Diese Funktion wird bereitgestellt, weil WARC-Dateien nicht größer als 1GB sein sollen und

[8] Hier und im Folgenden verwenden wir die Begriffe Website und Webpage gemäß der Definitionen des W3C. Eine Webpage ist demnach eine “collection of information, consisting of one or more Web resources, intended to be rendered simultaneously, and identified by a single URI” und eine Website ist definiert als “collection of interlinked Web pages, including a host page, residing at the same network location.” (Lavoie und Nielsen 1999)

[9] Webrecorder Project; Webrecorder Project (2021)

[10] [7]

[11] Diese automatisch erstellten Visualisierungen dienen nur zum ersten Eindruck; soll eine Visualisierung erstellt werden, die nicht nur für die Exploration, sondern für die Verwendung als Demonstration und Argumentation gedacht ist, bedarf es einer weitergehenden begründeten Auswahl eines angemessenen Visualisierungsalgorithmus.

Webseiten daher bei der Archivierung auf mehrere WARC-Dateien aufgeteilt werden können. Neben der Analyse von WARC-Dateien können auch Links zu Webpages im live web übergeben und damit nicht-archivierte Websites analysiert werden. Wird das Tool in Form der importierten Python Bibliothek Warc2graph verwendet, eröffnen sich weitere Anwendungsmöglichkeiten. Die Bibliothek stellt die Funktion `create_model` zur Verfügung, die parallel zur Kommandozeilenanwendung einen Pfad zu einer WARC-Datei benötigt. Die Funktion gibt einen gerichteten Graphen zurück, der mithilfe der von NetworkX[12], einer Python-Bibliothek für Graphdaten- und Netzwerkanalyse implementiert ist. Der breite Funktionsumfang von NetworkX kann nun direkt für den erstellten Graphen genutzt werden, um zum Beispiel Zentralitätsmaße zu berechnen oder die Zirkularität und andere spezifische Eigenschaften des Graphen zu prüfen.

3.2 Funktionsweise

Die Modellierung einer Website als Netzwerkgraph mithilfe von Warc2graph läuft in zwei Schritten ab. Die Funktion `warc2graph.extract_links` verarbeitet die WARC -Dateien, liest sie aus und extrahiert daraus alle Verweise zwischen Ressourcen, während die Funktion `warc2graph.create_network` aus den extrahierten Informationen direktionale Netzwerke erstellt. Nutzt man das Python-Modul können beide Schritte unabhängig voneinander durchgeführt und die Daten nach jedem Zwischenschritt überprüft und manuell angepasst werden. Um die Verweise aus der WARC-Datei zu extrahieren wird über alle Einträge der WARC-Datei iteriert. Jede HTML-Datei wird dann mithilfe von drei möglichen Extraktionsmethoden analysiert.

1. Die einfachste Methode liest die in der WARC-Datei gespeicherten Metadaten aus. In den Metadaten können die beim Crawling durch das Webarchivierungstool gefundenen ausgehenden Links (Outlinks) einer Ressource vermerkt sein. Diese Methode ist sehr robust und performant, gleichzeitig aber am wenigsten flexibel. Einzelne Domains können zwar gefilter werden, aber was beim Crawling nicht gefunden wurde kann auch jetzt nicht mehr gefunden werden. Hinzu kommt, dass die WARC-Spezifikationen nicht vorgeben, dass Webarchivierungstools beim Erstellen der WARC-Dateien Metadaten mit Outlinks anlegen müssen. Ob eine WARC-Datei Metadaten mit gefundenen Outlinks einer Ressource enthält, liegt also daran, mit welchem Tool sie erstellt wurde.

2. Bei einer weiteren Methode werden alle HTML-Dateien ausgelesen und mithilfe der Python-Bibliothek BeautifulSoup[13] ausgewertet. Hierbei können alle Tags, die im HTML vorliegen, gefunden werden. Links, die erst durch Javascript generiert werden, können hiermit aber nicht gefunden werden, da das Javascript nicht ausgewertet wird.

[12] [8]
[13] [17]

3. Mit dem Ziel, auch das Javascript auszuwerten, werden die HTML-Dateien mit einem code-gesteuerten Browser (Selenium[14] mit Firefox/Geckodriver) geöffnet und das dabei erstellte Document Object Model (DOM) verwendet, um sowohl die im HTML-Quelltext vorhandenen als auch die dynamisch durch Javascript generierten Links erkennen zu können. Die Steuerung des Browsers und die Auswertung des Javascript macht sich deutlich in der Laufzeit bemerkbar.

Die Informationen werden nach diesem Teilschritt vorerst in einer Liste gespeichert, die aus Tuples mit URLs der Ausgangsressource und URLs der Zielressource bestehen. Im folgenden Schritt wird die erstellte Liste mithilfe des Python Pakets NetworkX in einen Netzwerkgraphen umgewandet, bei dem jede Ressource – identifiziert über ihre URL – als Knoten und jeder Verweis als Kante von der Ausgangsressource zur Zielressource modelliert wird. Informationen über Zeitpunkt der Erstellung des Graphen und über dabei verwendete Parameter werden als Attribut des Graphen gespeichert. Die Daten und Metadaten des gesamten Graphen, aber auch jedes einzelnen Knotens können beliebig erweitert werden.

Der hier beschriebene Prozess von Warc2graph ist modular aufgebaut. Wer nur an einer Extraktion der Verweise, nicht aber an einem Netzwerkgraphen interessiert ist, kann sich die Liste der extrahierten Links ausgeben lassen. Wer Informationen aus anderen Quellen in einen Netzwerkgraphen umwandeln möchte, kann eine Liste aller möglichen Verweise manuell erstellen und diese von Warc2graph in einen Graphen umwandeln lassen, der die selbe Struktur hat, wie die aus einer WARC-Datei erstellten Graphen.

3.3 Reproduzierbarkeit der Ergebnisse

Warc2graph kann auch, per Übergabe einer URL, Websites im live web analysieren. Die Ergebnisse wären allerdings zu einem späteren Zeitpunkt, wenn die Seiten sich verändert haben oder offline sind, nicht reproduzierbar. Eine notwendige Voraussetzung für die vergleichende corpusorientierte Mustererkennung ist aber die Vergleichbarkeit und Reproduzierbarkeit der Einzelanalysen. Weil auch netzliterarische Webseiten sich mit der Zeit ändern können, muss bei wiederholbaren und vergleichenden Analysen mit archivierten Versionen der Seiten gearbeitet werden.

Der hohe Grad an unkontrollierten Veränderungen von Webseiten im live web führt dazu, dass Forschungsergebnisse, die auf extrahierten Daten basieren und nur die extrahierten Daten, nicht aber die Datenbasis verfügbar halten, nach kurzer Zeit nicht mehr reproduziert werden können. Die Extraktion von Verweisstrukturen aus WARC-Dateien ist dagegen dauerhaft reproduzierbar, weil die Datenbasis als Teil der Forschungsdaten mit archiviert werden kann.

Neben der Reproduzierbarkeit von Forschungsergebnissen erlaubt dies auch eine methodisch kontrollierte Verbesserung der eingesetzten Methoden und damit perspektivisch eine Verbesserung der Analyseergebnisse, in diesem Fall der Extraktion von Verweisstrukturen.

[14] Selenium Project

Methodisch kann dieser WARC-Workflow auch auf andere Ansätze übertragen werden, die ihre Daten durch die computergestützte Verarbeitung von von Webseiten gewinnen.

4 SDC4Lit Architektur

4.1 Aufbau und Struktur

Die Aufgabe des Projekts Science Data Center for Literature (SDC4Lit), das den institutionellen Rahmen für die Studie zur narrativen Struktur von Netzliteratur und die Entwicklung von Warc2graph bildet, ist die Realisierung eines nachhaltigen Datenlebenszyklus für Literaturforschung und -vermittlung zu Born-digital Materialien. Hierzu wird eine Forschungsumgebung aufgebaut, die den Zugang zu den archivierten digitalen Objekten sowie die Nutzung ausgewählter Analysemethoden und-werkzeuge für die Forschung ermöglicht. Die konzipierte SDC4Lit-Architektur besteht aus einem Primärdaten- und einem Forschungsdatenrepositorium mit einer zusätzlichen Analyseschicht und einem übergreifenden Portal.[15]

Im Primärdatenrepositorium werden digitale Objekte aus den Bereichen Literatur im Netz und den Vor- und Nachlässen des Deutschen Literaturarchivs in Marbach (DLA) langfristig gespeichert. Die Aufbereitung, Einarbeitung und Bereitstellung der Daten erfolgt durch Mitarbeiter*innen am DLA. Die geplante Analyseschicht stellt Methoden und Werkzeuge für die computergestützte Arbeit mit den digitalen Archivalien bereit. Zugehörige Forschungsdaten und Forschungsergebnisse von Nutzer*innen werden in einem separaten Forschungsdatenrepositorium gespeichert und können gegebenenfalls für die Nachnutzung zur Verfügung gestellt werden. Um die Bedarfe der Forschungscommunity bei der Entwicklung zu berücksichtigen werden im Rahmen des Projekts wissenschaftliche Fallstudien zu den bereitgestellten Primärmaterialien durchgeführt. Mit Blick auf Textumgangsformen im Bereich Elektronische Literatur und Born-Digitals werden auf diese Weise die Anforderungen an das Portal durch die Abbildung konkreter Arbeitsschritte informiert, wie sie etwa die Extraktion und Analye von Referenznetzwerken für Objekte im Bereich Literatur im Netz darstellt. Das übergreifende SDC4Lit-Portal soll die einzelnen Komponenten miteinander verbinden und den verschiedenen Nutzergruppen (Archiv, Forschung, Lehre, Publikum) einen Zugang zum Archivbestand gewähren. Eine Anbindung externen Repositorien ist geplant.

4.2 Dataverse als Repositoriumssoftware

Die Kernkomponenten des Portals bilden die Repositorien, für deren Aufbau eine passende Repositoriumssoftware ausgewählt sowie ein Datenmodell entwickelt werden muss.

[15] Weitere Informationen und die SDC4Lit Architektur Skizze sind im Extended Poster Abstract „SDC4Lit – Science Data Center for Literature“, der auch im Tagungsband der EST-21 Konferenz veröffentlicht wird, zu finden.

Angesichts der besonderen Form der Primärdaten stellen sich spezifische Anforderungen an das aufzubauende Repositorium. Hier war insbesondere die Herausforderung, dass die Primärdaten in Verzeichnisstrukturen organisiert sind. Da diese für die Forschung auch erhalten werden müssen, ergibt sich daraus die Anforderung, dass das Repositorium auch in der Lage sein muss, mit Verzeichnisstrukturen zumindest umzugehen. Die Anforderungen konnten nicht vollständig von den verfügbaren Softwarelösungen erfüllt werden, sodass hier voraussichtlich Eigenentwicklungen nötig sind. Anhand der ermittelten Anforderungen wurden mehrere Softwarelösungen analysiert und geprüft. Die Entscheidung fiel auf Dataverse[16]. In Dataverse lassen sich Verzeichnisstrukturen über das Metadatenfeld „FilePath“ nachbilden, darüber hinaus verfügt es über eine Große Nutzer*innen-Community, wird stetig fortentwickelt und hat trotzdem Produktcharakter. Darüber hinaus sind in SDC4Lit Erfahrungen aus dem bereits abgeschlossenen DIPL-ING-Projekt vorhanden, mit dem der Aufbau eines institutionellen Forschungsdatenrepositorium für die Universität Stuttgart mit Dataverse geleistet wurde.[19]

Dabei basiert eine Installation von Dataverse aus mehreren logischen Dataverses[17], die eine oberste Strukturierungsschicht darstellen. Im Fall von SDC4Lit werden die drei logischen Dataverses „Literatur im Netz“, „Born digitals“ und „Literarische Computerspiele“ als oberste Strukturierungsebene verwendet, unter die weitere logische Dataverses weitere Strukturierungsebenen einführen können. Am unteren Ende sind die Primärdaten selbst jeweils in Datasets[18] organisiert. Diese bestehen aus den Dateien, zu denen auch jeweils die Verzeichnisinformation als FilePath mit angegeben werden kann, sodass die oben genannte Anforderung erfüllt wird. Auf technologischer Ebene ist die Dataverse-Installation in einer virtuellen Maschine untergebracht, um leichte Migration und Imaging zu erlauben. Die Daten sind auf einer RAID5-Festplattenverbund verortet, um die Datensicherheit zu erhöhen. Darüber hinaus werden derzeit Backup-Routinen entworfen, um die Daten auf Bandspeicher zu sichern und in ein Langzeitarchiv zu überführen.

4.3 Datenmodellierung

Selbstverständlich werden Metadaten benötigt, damit die Forschungsdaten den FAIR-Prinzipien entsprechen [24]. Hier werden bei Dateverse Metadatenblöcke angelegt, die jeweils logisch zusammengehörende Metadaten darstellen und als Felder ausfüllbar sind. Diese Felder müssen von den Administratorinnen und Administratoren als TSV-Dateien eingepflegt werden und richten sich nach einem im Projektrahmen ausgewähltem oder definierten Standard. SDC4Lit entwickelt ein eigenes Datenmodell. Dabei wurden Metadatenstandards, die im Bereich des Bibliotheks- und Archivwesens gängig sind, wie METS [2],MODS [6] und PREMIS [3], evaluiert und ausgewählt. Metadaten, die bereits im Laufe der Sammlung, Archivierung und Erschließung entstehen, werden zusammen mit Primär-

[16]Dataverse Projekthomepage: `https://dataverse.org/`, aufgerufen: 28.04.2021.

[17]Dataverse User Guide: Dataverse Collection Management, `https://guides.dataverse.org/en/latest/user/dataverse-management.html?highlight=dataverse%20management`, aufgerufen: 28.04.2021.

[18]Siehe ebenda (Fußnote 14).

daten in Primärdatenrepositorium gespeichert, während Forschungsdaten und Metadaten, die im Laufe der Forschung entstehen, im parallel aufgebauten Forschungsdatenrepositorium gespeichert werden. Dataverse vergibt den Daten DOIs[19] worüber zusammengehörende Primär- und Forschungsdaten aufeinander referenziert werden können.

4.4 Einbindung von warc2graph in die Infrastruktur

Warc2graph wird als Teil der Analyseschicht in die SDC4Lit Infrastruktur eingebunden. Das Modul wird Input aus dem Primärdatenrepositorium und aus dem Forschungsdatenrepositorium beziehen und Output in das Forschungsdatenrepositorium schreiben können. Die generierten Outputs, die im Forschungsdatenrepositorium liegen, verweisen mittels DOI auf die zugrundeliegenden Primärdaten, und umgekehrt. Für alle Netzliteraturobjekte wird Warc2graph angewendet, um sie mit den Ergebnissen anreichern zu können, und damit einen besseren Zugang zu ermöglichen. Leser*innen können auf einen Blick sehen, ob das Werk wenige oder viele Ressourcen umfasst, mit welchen HTML-Tags die Ressourcen referenziert sind und welche Struktur sich daraus ergibt. In Verbindung mit einem Replay der WARC-Objekte sind diese grundlegenden Sturkturinformationen sowohl für literaturwissenschaftliche als auch für erhaltungsbezogene Forschung relevant sein.

5 Nachnutzung

Die von uns präsentierten Modelle, Workflows und die zugehörige Software wurden anhand eines spezifischen WARC-Korpus entwickelt. Das bedeutet, das bestimmte analytische Verfahrensweisen und die Leistung der Software abhängig ist von den Eigenschaften des Korpus. Gleichzeitig kann durch den hohen Standardisierungsgrad des WARC-Formats eine hohe strukturelle Ähnlichkeit des verwendeten Korpus mit anderen WARC-Korpora vorausgesetzt werden. Die Analyse von Referenznetzwerken in WARC-Korpora ist anschlussfähig für ganz unterschiedliche Fragestellungen, weil die Extraktion der Referenzen und die Transformation der WARC-Daten in ein graphbasiertes Datenformat bei der Korpusanalyse relativ weit am Anfang von analytischen Workflows angesiedelt sind. Wir gehen davon aus, dass die Extraktion von Referenzen generisch für die Verarbeitung von WARC-Korpora eingesetzt werden kann. Aufgrund des spezifischen Entwicklungskorpus ist eine fallbezogene Überprüfung der Extraktionsergebnisse geboten. Eine generische oder zumindest methodisch kontrollierte Nachnutzung ist erst nach Durchführung eines systematischen Qualitäts- und Leistungstests anhand ausgewählter Testkorpora denkbar. Ein solcher Test steht noch aus.

Die Transformation von WARC-Dateien in ein graphbasiertes Datenformat eröffnet über die Weiterverarbeitung der Referenzstrukturen hinaus weitergehende Möglichkeiten der Archivierung und der Analyse, insofern insbesondere hypermediale Objekte zunächst in medienspezifische Elemente aufgeteilt werden. Diese Aufteilung ermöglicht dann medien-

[19] Digital Objekt Identifier System: `https://www.doi.org/`, aufgerufen: 28.04.2021.

bzw. formatspezifische Zusammenstellungen und Analysen, z.B. Textanalysen oder Bildanalysen sowie graphbasierte Analyseansätze. Im hier gewählten Ansatz führt die korpusbasierte Modellierung, Entwicklung und Durchführung der Verweisextraktion zu Daten, die im Forschungsdatenrepositorium strukturiert vorgehalten werden und für die Nachnutzung freigegeben sind. Für den archivarischen Umgang mit den Daten ist entscheidend, dass die Referenzstrukturen und Graphdaten als grundlegende Strukturanalyse in der Regel einer frühen Phase von Forschungsprozessen zugeordnet und daher für weitere fachspezifische Forschungsfragen anschlussfähig sind - etwa für genaue Lektüren einzelner Werke mit Blick auf nicht-lineare Erzählstrukturen.

6 Acknowledgements

Das Science Data Center for Literature wird finanziert vom Ministerium für Wissenschaft und Kultur Baden Württemberg. Am Projekt beteiligt sind das Deutsche Literaturarchiv Marbach, das Höchstleistungsrechenzentrum der Universität Stuttgart sowie das Institut für Maschinelle Sprachverarbeitung und das Institut für Literaturwissenschaft der Universität Stuttgart.

Literaturverzeichnis

[1] Berkenheger, Susanne. 1997. Zeit für die Bombe. Hyperfiction. `http://www.berkenheger.netzliteratur.net/ouargla/wargla/zeit.htm` (zugegriffen: 8. Mai 2021).

[2] Cantara, Linda. 2005. METS: The Metadata Encoding and Transmission Standard. *Cataloging & Classification Quarterly* 40, Nr. 3-4 (September): 237–253. doi: `https://doi.org/10.1300/J104v40n03\_11` (zugegriffen: 10. Mai 2021).

[3] Caplan, Priscilla. 2009. Understanding PREMIS: an overview of the PREMIS Data Dictionary for Preservation Metadata. Library of Congress.

[4] Eldakar, Youssef und Lana Alsabbagh. 2020. LinkGate: Let's build a scalable visualization tool for web archive research. April. `https://netpreserveblog.wordpress.com/2020/04/23/linkgate-update/` (zugegriffen: 8. Mai 2021).

[5] Ensslin, Astrid. 2007. *Canonizing Hypertext : Explorations and Constructions*. London: Continuum International Publishing.

[6] Gartner, Richard. 2003. MODS: Metadata Object DescriptionSchema. JISC Techwatch report TSW.

[7] GEXF Working Group. 2009. GEXF File Format. `https://gephi.org/gexf/format/` (zugegriffen: 29. April 2021).

[8] Hagberg, Aric A., Daniel A. Schult und Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In: *Proceedings of the 7th Python in Science Conference*, hg. von Gaël Varoquaux, Travis Vaught, und Jarrod Millman, 11–15. Pasadena, CA USA.

[9] Hein, Pascal, Mona Ulrich, Claus-Michael Schlesinger und André Blessing. 2021. warc2graph. Zenodo, Mai. `https://zenodo.org/record/4742254` (zugegriffen: 8. Mai 2021).

[10] IIPC. The WARC Format. `https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/` (zugegriffen: 7. Mai 2021).

[11] Landow, George P. 1997. *Hypertext 2.0 : Hypertext - the convergence of contemporary critical theory and technology*. Rev., amplified ed. Baltimore, Md. [u.a.]: Johns Hopkins Univ. Press.

[12] Lavoie, Brian und Henrik Frystyk Nielsen. 1999. Web Characterization Terminology & Definitions Sheet. *Web Characterization Terminology & Definitions Sheet*. `https://www.w3.org/1999/05/WCA-terms/` (zugegriffen: 29. April 2021).

[13] Lin, Jimmy, Ian Milligan, Jeremy Wiebe und Alice Zhou. 2017. Warcbase: Scalable Analytics Infrastructure for Exploring Web Archives. *J. Comput. Cult. Herit.* 10, Nr. 4 (Juli): 22:1–22:30. doi:`http://doi.acm.org/10.1145/3097570` (zugegriffen: 19. November 2019).

[14] Marbach, Deutsches Literaturarchiv. 2018. Literatur im Netz. `http://literatur-im-netz.dla-marbach.de/` (zugegriffen: 7. Mai 2021).

[15] Martinez, Matias und Michael Scheffel. 2007. *Einführung in die Erzähltheorie*. München: C.H. Beck.

[16] Mike Burner und Brewster Kahle. 1996. Arc File Format. *Internet Archive: ARC File Format Reference*. September. `https://archive.org/web/researcher/ArcFileFormat.php` (zugegriffen: 10. Mai 2021).

[17] Richardson, Leonard. 2020. Beautiful Soup Documentation. `https://www.crummy.com/software/BeautifulSoup/bs4/doc/` (zugegriffen: 29. April 2021).

[18] Selenium Project. The Selenium Browser Automation Project. `https://www.selenium.dev/documentation/en/` (zugegriffen: 29. April 2021).

[19] Selent, Björn, Björn Schembera, Dorothea Iglezakis und Anett Seeland. 2019. Datenmanagement in Infrastrukturen, Prozessen und Lebenszyklen für die Ingenieurwissenschaften : Abschlussbericht des BMBF-Projektes Dipl-Ing. Universität Stuttgart. doi:10.2314/KXP:1693393980, `https://www.tib.eu/suchen/id/TIBKAT:1693393980/` (zugegriffen: 10. Mai 2021).

[20] Suter, Beat, Michael Böhler und Christian Bachmann, Hrsg. 1999. *Hyperfiction: hyperliterarisches Lesebuch: Internet und Literatur*. Nexus 50. Frankfurt am Main: Stroemfeld.

[21] Webrecorder Project. 2021. webrecorder/warcio. Webrecorder, Mai. https://github.com/webrecorder/warcio (zugegriffen: 5. Mai 2021).

[22] ---. Webrecorder pywb 2.5. *GitHub - webrecorder/pywb: Core Python Web Archiving Toolkit for replay and recording of web archives.* https://github.com/webrecorder/pywb (zugegriffen: 10. Mai 2021a).

[23] ---. WARCIO: WARC (and ARC) Streaming Library. https://github.com/webrecorder/warcio (zugegriffen: 7. Mai 2021b).

[24] Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, u. a. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, Nr. 1 (Dezember): 160018. doi:10.1038/sdata.2016.18, http://www.nature.com/articles/sdata201618 (zugegriffen: 10. Mai 2021).