

Thomas Efer

2017

Graphdatenbanken für die textorientierten e-Humanities

Alternative Textrepräsentation
für neuartige Rechtersysteme
und Text-Mining-Verfahren



Eigenverlag

Informatik

Bibliographische Beschreibung

- Titel:* Graphdatenbanken für die textorientierten e-Humanities
- Autor:* Thomas Efer
<http://d-nb.info/gnd/1125649186>
<http://orcid.org/0000-0002-8376-3884>
- Ausgabe:* 18. August 2017
- Fachgebiet:* Informatik
- Sprache:* Deutsch
- Umfang:* 194 Seiten Hauptteil, 45 Abbildungen, 4 Tabellen, 6 Quelltexte
- Schlagwörter:* Graphdatenbanken, Datenmodellierung, Recherchesysteme, e-Humanities, Text Mining, Korpusexploration, Information Retrieval
- Notiz:* Dieses Buch enthält eine leicht überarbeitete Version der gleichnamigen Dissertationsschrift, welche am 15. Februar 2017 auf dem sächsischen Dokumentenserver unter der Adresse <http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa-219122> veröffentlicht wurde. Für wissenschaftliche Zitationen ist bevorzugt die Originalfassung zu verwenden.

Abstract

English Version

In light of the recent massive digitization efforts, most of the humanities disciplines are currently undergoing a fundamental transition towards the widespread application of digital methods. In between those traditional scholarly fields and computer science exists a methodological and communicational gap, that the so-called "e-Humanities" aim to bridge systematically, via interdisciplinary project work. With text being the most common object of study in this field, many approaches from the area of Text Mining have been adapted to problems of the disciplines. While common workflows and best practices slowly emerge, it is evident that generic solutions are no ultimate fit for many specific application scenarios. To be able to create custom-tailored digital tools, one of the central issues is to digitally represent the text, as well as its many contexts and related objects of interest in an adequate manner.

This book introduces a novel form of text representation that is based on Property Graph databases – an emerging technology that is used to store and query highly interconnected data sets. Based on this modeling paradigm, a new text research system called "Kadmos" is introduced. It provides user-definable asynchronous web services and is built to allow for a flexible extension of the data model and system functionality within a prototype-driven development process. With Kadmos it is possible to easily scale up to text collections containing hundreds of millions of words on a single device and even further when using a machine cluster. It is shown how various methods of Text Mining can be implemented with and adapted for the graph representation at a very fine granularity level, allowing the creation of fitting digital tools for different aspects of scholarly work. In extended usage scenarios it is demonstrated how the graph-based modeling of domain data can be beneficial even in research scenarios that go beyond a purely text-based study.

Deutsche Version

Vor dem Hintergrund zahlreicher Digitalisierungsinitiativen befinden sich weite Teile der Geistes- und Sozialwissenschaften derzeit in einer Transition hin zur großflächigen Anwendung digitaler Methoden. Zwischen den Fachdisziplinen und der Informatik zeigen sich große Differenzen in der Methodik und bei der gemeinsamen Kommunikation. Diese durch interdisziplinäre Projektarbeit zu überbrücken, ist das zentrale Anliegen der sogenannten „*e-Humanities*“. Da Text der häufigste Untersuchungsgegenstand in diesem Feld ist, wurden bereits viele Verfahren des *Text Mining* auf Problemstellungen der Fächer angepasst und angewendet. Während sich langsam generelle Arbeitsabläufe und *Best Practices* etablieren, zeigt sich, dass generische Lösungen für spezifische Teilprobleme oftmals nicht geeignet sind. Um für diese Anwendungsfälle maßgeschneiderte digitale Werkzeuge erstellen zu können, ist eines der Kernprobleme die adäquate digitale Repräsentation von Text sowie seinen vielen Kontexten und Bezügen.

In diesem Buch wird eine neue Form der Textrepräsentation vorgestellt, die auf Property-Graph-Datenbanken beruht – einer aktuellen Technologie für die Speicherung und Abfrage hochverknüpfter Daten. Darauf aufbauend wird das Textrecherchesystem „Kadmos“ vorgestellt, mit welchem nutzerdefinierte asynchrone Webservices erstellt werden können. Es bietet flexible Möglichkeiten zur Erweiterung des Datenmodells und der Programmfunktionalität und kann Textsammlungen mit mehreren hundert Millionen Wörtern auf einzelnen Rechnern und weitaus größere in Rechnerclustern speichern. Es wird gezeigt, wie verschiedene *Text-Mining*-Verfahren über diese Graphrepräsentation realisiert und an sie angepasst werden können. Die feine Granularität der Zugriffsebene erlaubt die Erstellung passender Werkzeuge für spezifische fachwissenschaftliche Anwendungen. Zusätzlich wird demonstriert, wie die graphbasierte Modellierung auch über die rein textorientierte Forschung hinaus gewinnbringend eingesetzt werden kann.

Inhaltsverzeichnis

Bibliographische Beschreibung	I
Abstract (Englisch und Deutsch)	II
Inhaltsverzeichnis	IV
1 Einleitung	2
1.1 Motivation	3
1.2 Forschungsfragen	9
1.3 Beiträge zum Forschungsfeld	10
1.4 Aufbau des Buches	13
2 Forschungskontext und relevante Technologien	14
2.1 e-Humanities	15
2.1.1 Entwicklung und Selbstverständnis des Fachgebiets	15
2.1.2 Forschungsmethodik und aktuelle Entwicklungen	20
2.1.3 Forschungsressourcen und -infrastrukturen	24
2.2 Text- und Korpusrepräsentation	28
2.2.1 Charakterisierung von Forschungskorpora	28
2.2.2 Zeichenrepräsentation	31
2.2.3 Repräsentation der Struktur von Text	34
2.2.4 Textrepräsentation im Text Mining	39
2.2.5 Dokumentrepräsentation	41
2.2.6 Repräsentation von Metadaten und Annotationen	43
2.3 Graphdatenbanken	46
2.3.1 NoSQL-Datenbanken	46
2.3.2 Netzwerke, Graphen und ihre Anwendungsgebiete	50
2.3.3 Formalisierung und graphentheoretische Zugänge	52
2.3.4 Property-Graph-Datenbanken	55

2.3.5	Semantic-Web-Technologien	58
2.3.6	Abfragesprachen	62
2.4	Vorarbeiten und verwandte Gebiete	64
2.5	Ableitbare Systemanforderungen	69
3	Kadmos – ein graphbasiertes Recherchesystem	72
3.1	Entwicklungsziele	73
3.2	Daten- und Domänenmodell	75
3.3	Technologie und Systemarchitektur	79
3.4	Asynchrone Webservicearchitektur	84
3.5	Datenimport	87
3.6	Zeichennormalisierung	91
3.7	Flexibles graphbasiertes Information Retrieval	95
3.7.1	Retrievalverfahren und Textrepräsentation	95
3.7.2	Facettierung über nutzerspezifische Metadaten	99
3.7.3	Von Schlagwörtern zur konzeptbasierten Suche	102
3.7.4	Ergebnisrepräsentation und Retrievalstrategien	107
3.8	Graphbasierte Korpusexploration	109
3.9	Evaluierung	114
3.10	Erweiterungsmöglichkeiten	122
3.10.1	Anlegen neuer Service-Endpunkte	122
3.10.2	Gekapselte Erweiterung mit dem Plugin-System	125
4	Modellerweiterungen und komplexere Anwendungsfälle	130
4.1	Erweiterbarkeit und Konstruktive Voraussicht	131
4.2	Entitäten-Netzwerke	134
4.2.1	Navigationsunterstützung und Netzwerkanalyse	134
4.2.2	Erzeugung und Exploration von Toponymnetzwerken	140
4.2.3	Eigennamenübersetzung aus lokal alignierten Paralleltexten	145
4.3	Systematisierung und Filterung bibliographischer Daten	149
4.4	Struktur und Komplexität von Dramen	160
4.4.1	Extraktion, Analyse und Visualisierung von Struktur	160
4.4.2	Informationstheoretische Komplexitätsbetrachtungen	166
4.5	Aufbau interner Indexstrukturen für lexikalische Ähnlichkeit	173
4.6	Das Graphenparadigma als interdisziplinäres Kommunikationsmittel	177

5 Schlussbetrachtungen	184
5.1 Zusammenfassung	185
5.2 Technologische und methodische Grenzen	187
5.3 Ausblick	190
Literaturverzeichnis	196
Abkürzungsverzeichnis	229
Abbildungsverzeichnis	238
Quelltextverzeichnis	240
Tabellenverzeichnis	241
A Verwendete Korpora	242
B Ergänzende Grafiken	246
C Anleitung zur Inbetriebnahme der Kadmos-Umgebung	251

Kapitel 1

Einleitung

*The world has arrived at an age of cheap
complex devices of great reliability;
and something is bound to come of it.*

Vannevar Bush

US-amerikanischer Ingenieur und Professor für Elektrotechnik

aus dem Essay „As We May Think“ (1945) – [[Bus45](#)]

1.1 Motivation

Die häufige Charakterisierung unserer heutigen Epoche als das „Digitale Zeitalter“ ist ein Beleg für den großen Einfluss und die zentrale Bedeutung digitaler Technologien für die gesamte Gesellschaft. Der Informatik als grundsätzlich eigenständiger und unabhängiger akademischer Disziplin kommt damit die immer wichtigere und mit großer Verantwortung verbundene Rolle zu, geeignete Rahmenbedingungen für effiziente Prozesse in zentralen Sektoren, wie Produktion, Handel und Logistik, Wissenschaft und Verwaltung, aber auch Kommunikation und Kultur zu schaffen.

Während die Informatik sich bereits seit langem mit Fragen von gesellschaftlicher Dimension auseinandersetzt – ab den 1950er Jahren etwa, mit den in der Kybernetik wurzelnden Überlegungen zur künstlichen Intelligenz – ist die praktische Anwendbarkeit solcher Forschungsfelder und die soziale Relevanz der Informatik erst in den letzten Jahren für die Breite der Gesellschaft greifbar geworden.

Die Geistes- und Sozialwissenschaften auf der anderen Seite, welche seit jeher die Rolle kritischer Betrachter sozialer Themen einnehmen, und deren Forschen und Wirken grundsätzlich als gesellschaftliches Korrektiv dienen soll, haben die rasanten Entwicklungen der Informationstechnologie bislang nur sehr zögerlich aufgegriffen. Viele Disziplinen, die über Jahrzehnte oder gar Jahrhunderte lang Forschungsmethodik tradiert und Theorien gepflegt haben, stehen im Zuge des „digitalen Wandels“ heute vor einem grundlegenden Transformationsprozess. Egal, ob ein Fachgebiet der Adaption digitaler Methoden offen oder eher ablehnend gegenübersteht, so wandeln sich doch in jedem Fall Umfeld und Bedingungen der Forschung, wie auch die Erwartungen an sie.

Ohne diese allgemeine Entwicklung grundsätzlich werten zu wollen, lässt sich feststellen, dass die voranschreitende Digitalisierung einige unbestreitbare Vorteile mit sich bringt und einmalige Chancen für die akademische Welt insgesamt eröffnet. Durch die zahlreichen Möglichkeiten der digitalen Kommunikation zwischen Individuen und Gruppen wird eine globale Kollaboration innerhalb von Forschungs-Communities und über Disziplinen-Grenzen hinweg gefördert. Es bieten sich direktere Reaktionsmöglichkeiten auf neue Trends und Strömungen, wodurch nicht zuletzt auch der wissenschaftliche Diskurs angeregt wird. Dieser wird zudem durch neue Kommunikationsformate und -plattformen gleichzeitig öffentlicher und transparenter.¹

¹vgl. etwa [Nie11] und [SS13]

Die technologischen Fortschritte und veränderten Kommunikationswege läuten überdies eine Liberalisierung des wissenschaftlichen Publikationswesens ein. Die enorm gesunkenen Aufwände für das Veröffentlichen und Verbreiten von Inhalten sorgen für ein stark erhöhtes Publikationsaufkommen, besonders im Bereich digitaler Journals. Dadurch wird ein Pluralismus divergenter Meinungen gefördert und eine angemessene Abdeckung von Nischenthemen erstmals möglich. Die Schattenseite dieser Entwicklung ist die zunehmende Unsicherheit über die Verantwortlichkeiten und konkreten Durchführungsbedingungen der wissenschaftlichen Qualitätskontrolle für die publizierten Inhalte. Die schon immer schwierige Position traditioneller Verlagshäuser zwischen monetären Eigeninteressen und ihrer wichtigen Rolle als externe Überwacher und Lenker des akademischen Wissensaustauschs wird angesichts immer neuer digitaler Konkurrenz verschärft. Aktuelle Diskussionen im Zuge der Hinwendung zum Prinzip des *Open Access* betrachten bislang hauptsächlich Fragen des möglichst breiten Zugangs zu wissenschaftlichen Veröffentlichungen. In naher Zukunft wird sich auch die Suche nach geeigneten Konzepten für die (Selbst-)Kontrolle der wissenschaftlichen Qualität in diese Überlegungen einreihen müssen.²

Gleichzeitig zu diesen Umwälzungen in den organisatorischen Rahmenbedingungen für wissenschaftliche Publikationen entwickeln sich durch die Nutzung neuer Medien auch zahlreiche weitergreifende Möglichkeiten für die Dokumentation des Forschungsprozesses, etwa über das Bereitstellen umfangreicher (und teils interaktiv nutzbarer) Zusatzmaterialien für Fachveröffentlichungen. Durch das Publizieren von Forschungsdaten und Forschungssoftware wird das Verständnis der Herleitung neuer Erkenntnisse gefördert, das Ableiten alternativer Interpretationen erleichtert sowie eine Reproduzierbarkeit und objektivere Vergleichbarkeit von Forschungsarbeiten in einem bisher nicht dagewesenen Detailgrad ermöglicht. Es etabliert sich zudem langsam ein neuer Modus der Kollaboration durch die Nutzung von Forschungsinfrastrukturen, wo verwendete Daten und Verfahren gesichert, ausgetauscht, veröffentlicht und (ggf. in ganz unerwarteten Kontexten) nachgenutzt werden können.

Während die neuen Publikations- und Kollaborationsmöglichkeiten für sich genommen bereits großen Einfluss auf die Wissenschaftswelt ausüben, so zeigen sich die größten Auswirkungen des technologischen Wandels auf viel grundlegenderer Ebene: in der Art und Weise, wie geforscht wird. Im Hinblick auf die quellenorientierte Forschung ist die

²vgl. z. B. [Cav12] und [Ris14] und s. auch entsprechende Überlegungen aus dem „WRoTe Digital Humanities Dialog“, 2013, <http://dialog.e-humanities.net/assets/dialog/wrote-2013/Gruppendiskussion.pdf>

Adaption einer digitalen Arbeitsweise im Forschungsprozess immer dann unausweichlich, wenn die Analyse großer Quellenbestände nötig wird, sowie, wenn eine umfassendere und systematisierte Sicht auf bisher traditionell ausgewertete Primärquellen und die dabei entstandene Sekundärliteratur möglich werden soll.

Wichtigste Triebfeder dieses digitalen Wandels in den quellenorientierten Geistes- und Sozialwissenschaften ist die derzeit großflächig durchgeführte Digitalisierung analoger Bestände in Bibliotheken, Museen, Stiftungen und Archiven. Momentan ist diese primär motiviert durch konservatorische Überlegungen zur dauerhaften Bewahrung kulturellen Erbes und mithin des wertvollen Wissens, das bisher nur physisch auf, in, und in Form von (teils fragilen) Objekten hinterlegt ist³. Daran ist jedoch oft auch der Wunsch einer besseren Kategorisierung, Verwaltung und Beschreibung der eigenen Bestände über digitale Methoden verknüpft. Zudem wird bei Digitalisierungsinitiativen (besonders, wenn sie aus Mitteln der öffentlichen Hand finanziert werden) häufig auch die Wissenschaft als Hauptnutzer der entstehenden digitalisierten Sammlungen adressiert. In solchen erweiterten Nutzungsszenarien muss die Konzeption und Durchführung der Digitalisierungsarbeiten entsprechend qualifiziert ablaufen, um eine Eignung für die jeweils avisierte Form der Weiterverarbeitung zu gewährleisten.

Die [Deutsche Forschungsgemeinschaft \(DFG\)](#) hat mit [\[DFG13\]](#) entsprechende Empfehlungen erstellt, in denen Regeln und Richtwerte für die digitale Erfassung von Objekten (primär als Bild) und die Anreicherung dieser Digitalisate mit den zugehörigen Zusatzinformationen (so genannten Metadaten, durch welche der entsprechende Datensatz auffindbar gemacht werden soll) enthalten sind.

Mit dem Hinzufügen immer weiterer und feingliedriger Metadaten verschiebt sich auch der Fokus vom reinen Erhalt von Sammlungen hin zu ihrer digitalen Erschließung. Simple Katalogerfassung wird dabei durch subjektivere Vorgänge und fachwissenschaftliche Interpretation ergänzt, durch welche der Digitalisierungsprozess nah an die Forschung selbst heranrückt. Auch hier stellt sich die Frage nach einem angemessenen Umgang mit den sich bietenden Möglichkeiten. Besonders die historisch arbeitenden Wissenschaften sehen sich mit einem plötzlich massiv auftretenden Bedarf an wissenschaftlicher Erschließung großer Bestände konfrontiert. Der dabei zu beobachtende Paradigmenwechsel wird in [\[Mic16\]](#) treffend beschrieben:

³In dieser Betrachtung werden inhärent „oberflächliche“ *Mass-Digitization*-Projekte aus dem kommerziellen Umfeld (z. B. *Google Book Search*) wegen ihres grundsätzlich umstrittenen Nutzens für die Wissenschaft (vgl. [\[GTW13\]](#) für eine detaillierte Analyse) zunächst ausgeklammert.

[...] the advent of the digital library and other digital resources has profoundly altered the historian's relationship to information. The common prior condition of information scarcity has given way to overabundance.

Der Umgang mit einem plötzlichen Überfluss an Information ist für die Gelehrtenwelt nichts fundamental Neues,⁴ doch das derzeitige Ausmaß angesichts der relativ kurzen Zeitspanne, seit der die Digitalisierung Einzug in die Fachgebiete hält, macht ihn weit weniger beherrschbar⁵. Ganz allgemein gesehen fehlt in der Wissenschaft – wie auch in anderen zentralen Sektoren – vielfach die nötige Digitalkompetenz. Eine qualifizierte Herangehensweise an die Lösung von Problemen mit digitalen Mitteln hat sich noch nicht als Kulturtechnik etabliert. Entsprechend werden derzeit auch in den Geisteswissenschaften verstärkt Experten benötigt, die entsprechende Werkzeuge bereitstellen können.

Hier ergibt sich ein starker Berührungspunkt zur traditionell eher naturwissenschaftlich geprägten Technikwelt und zur Informatik. Während frühere Vorhaben zur Adaption digitaler Werkzeuge, wie Datenbanken, Katalogsysteme und Statistik-Software teils noch autark innerhalb der Fachdisziplinen umgesetzt werden konnten, stoßen diese Ansätze nun langsam an ihre Grenzen. Die Notwendigkeit zur interdisziplinären Arbeit eröffnet derzeit für alle Seiten neue Chancen für Kollaborationen jenseits des eigenen Fachgebiets. Aufgrund geringer inhaltlicher Schnittmengen und fundamental unterschiedlicher Wissenschaftstraditionen nehmen die Fachwissenschaften und die Informatik jedoch in gemeinsamen Forschungsvorhaben naturgemäß die Rollen von Antagonisten ein, welche bestrebt sind, den Entwicklungsprozess vom eigenen Standpunkt her zu beschreiben und zu lenken. Daraus ergibt sich nicht selten Konfliktpotential, insbesondere, wenn damit eine Reduzierung der Informatik auf einen reinen Technik-Dienstleister oder umgekehrt die technologieinduzierte Beschränkung des Einflusses der Fachwissenschaften auf eine simple massenhafte Dateneingabe verbunden ist.

Um diese Hürden zu umschiffen, widmen sich die *e-Humanities* bzw. *Digital Humanities* einer gemeinsamen Erarbeitung von Forschungsmethodik für ein angenehmeres, produktiveres, zielgerichtetes und adäquates Arbeiten in diesem emergenten Feld überlappender Forschungsinteressen. Die Aktivitäten setzen sowohl bei der Entwicklung allgemeiner

⁴siehe z. B. [Bla10] zum Einfluss der (subjektiv wahrgenommenen und objektiv belegbaren) Schwemme gedruckter Bücher ab dem beginnenden 16. Jahrhundert auf die europäischen Gelehrten und die Wissensproduktion

⁵An dieser Stelle soll zur Fokussierung der einleitenden Überlegungen nicht detaillierter auf die abzusehenden Probleme zukünftiger Historiker im Umgang mit den Artefakten unserer heutigen digitalen Wissensproduktion eingegangen werden, jedoch sei beispielhaft auf [Ros03] für eine kritische Betrachtung dieser Thematik verwiesen.

Theorien als auch beim anwendungsnahen Sammeln von „*Best Practices*“ an, wobei auch da die Arbeit an generischen und an spezialisierten Werkzeugen unterschieden werden kann.

In diese Prozesse können beide Seiten ihre jeweiligen Stärken einbringen. Wichtige Kompetenzen der Informatik liegen dabei u. a. in der formell korrekten und technisch effizienten Modellierung von Daten, im Umgang mit sehr großen Datenmengen, der Vernetzung von Wissensressourcen, der Entwicklung von Analyseverfahren und -werkzeugen sowie der Visualisierung von Analyseergebnissen. Die Geisteswissenschaften bringen dagegen u. a. sorgfältige manuelle Arbeitsweisen, theoriegeleitete Auswertungs- und Interpretations-Frameworks, ein hohes Verständnis für relevante Kontexte der Quellen und Daten sowie Erfahrung im Umgang mit unscharfen und unterspezifizierten Kategorien ein.

Während in diesem derzeit populären Bereich auch viele neue Ansätze entwickelt werden, befindet sich aus Sicht der Informatik (bis auf einige löbliche Ausnahmen) die Großzahl der bisher in den Geisteswissenschaften breit eingesetzten digitalen Werkzeuge im Hinblick auf Datenmodellierung, Datenhaltung, Abfragemöglichkeiten und Interaktionschnittstellen noch auf dem Forschungsstand der 1990er Jahre.

Um diesen Zustand nachhaltig zu verändern, bedarf es einer Analyse der aktuellen technologischen Entwicklung. Abseits der sozial- und geisteswissenschaftlichen Anwendungsdomäne (und teils auch ohne starke Verbindung zur Informatik-Forschung) prosperieren derzeit viele neue wissens- und datenverarbeitende Gebiete. Berufsbezeichnungen, wie „*Data Analyst*“ und „*Data Scientist*“ durchziehen die Stellenangebote in fast allen Branchen. Allgegenwärtige Datenquellen (in Unternehmen, öffentlichen Stellen und über offene digitale Kommunikationskanäle) machen Daten zu einem wichtigen und gut verfügbaren „Rohstoff“, dessen Verarbeitung und Erschließung⁶ sowie Auswertung Wettbewerbsvorteile verspricht.

Es werden hierbei Technologien benötigt, mit denen Wichtiges von Unwichtigem getrennt, der Grad der Strukturierung erhöht sowie der Kontext der Daten zunächst isoliert erfasst und anschließend durch eine Verknüpfung mit bereits Bekanntem erweitert werden kann. Das Auffinden und die Anreicherung großer Datensammlungen sind dabei an der Tagesordnung, so dass Schlagworte wie *Big Data* mittlerweile allgemeine Bekanntheit (und im Bereich personenbezogener Daten auch erste berechtigte Kritik) erlangt haben.

⁶passenderweise oft als *Data Mining* bezeichnet – bei textuellen Daten als *Text Mining*

Angesichts enorm großer Bestände von über das Internet verfügbaren, verknüpften Informationsressourcen und nicht zuletzt durch die wirkmächtige Popularisierung von „Sozialen Netzwerken“ wird begreiflich, dass auf gewisse Weise „alles mit allem vernetzt ist“⁷. Wird diese Beobachtung konsequent verinnerlicht, wird dadurch eine Auswertung von Daten in ihrem Kontext bzw. in ihren vielen Kontexten zum einen erst möglich und zum anderen dringend erforderlich.

Das vorliegende Buch greift sich im Folgenden mit der Fokussierung auf Graphdatenbanken eine Technologie aus der aktuellen Entwicklung heraus, die dieser Sichtweise in besonderem Maße gerecht werden kann und die ein aktueller Forschungsgegenstand der Informatik ist. Sie wird auf die Speicherung und Verarbeitung von Text, als der primären Quellenform der Geisteswissenschaften, erweitert. Auch Metadaten, lexikalisches Wissen und nutzerdefinierte Konzeptualisierungen werden damit abbildbar. Das System bildet die Basis für vielseitige Rechercheanwendungen.

Trotz prinzipiell technischer Ausrichtung des Buches und der grundsätzlich universellen Anwendbarkeit der vorgestellten Verfahren für textorientierte Recherchen wird bewusst der Bezug zu den Sozial- und Geisteswissenschaften hergestellt, da der Autor eine Annäherung auf digitalem Terrain als perspektivisch wichtig ansieht: Die Qualifizierung der Informatik in Richtung des tieferen Verständnisses gesellschaftlich relevanter Fragestellungen und dem adäquaten Umgang mit kulturellen Artefakten – insbesondere textuellen Quellen – muss sich ebenso vollziehen, wie die Qualifizierung der Geisteswissenschaften in Richtung der informierten und zweckmäßigen Nutzung aktueller Technologien.

Oft wurde in der Vergangenheit schon thematisiert, dass beiden Seiten ein breiteres Gespür für die relevanten Fragestellungen der jeweils anderen fehlt. Diesem Buch liegt die Hoffnung zugrunde, dass diese Hürden durch die gemeinsame, schrittweise und forschungsgeleitete Entwicklung von adäquaten Werkzeugen im Rahmen der e-Humanities abgebaut werden können. Letztendlich soll mit der Urbarmachung neuer und moderner technologischer Basiskomponenten eine Grundlage für eben solche gemeinschaftlichen Projekte geschaffen werden.

⁷vgl. [Bar02] als Ausdruck der „neuen Wissenschaft von Netzwerken“

1.2 Forschungsfragen

Im Angesicht der oben beschriebenen Ausgangslage kommen auf die Informatik viele neuartige Problemstellungen zu, die sowohl im Bereich der anwendungsbezogenen Forschung (speziell im interdisziplinären Kontext) angesiedelt sind, als auch Kernthemen der Informatik, wie Datenmodellierung und -repräsentation sowie Algorithmik betreffen. In dieser Arbeit soll nicht vorrangig ein konkretes Teilproblem fokussiert gelöst werden. Das wichtigere Ziel ist es, eine grundsätzliche, teils technologisch motivierte, teils von offenen Fachfragen der Geisteswissenschaften gelenkte Ergänzung bisheriger Problemlösungen und Lösungsstrategien zu ermöglichen. Um die breite Anwendbarkeit der vorgestellten Überlegungen und Lösungen zu sichern, wird im Laufe der Ausführungen unterschiedlichen Fragestellungen nachgegangen. Über diesen spezialisierten Aspekten stehen jedoch einige grundsätzliche Fragen, zu denen in diesem Buch Erkenntnisse gewonnen werden sollen.

Die untersuchte Technologie der Graphdatenbanken stellt einen vergleichsweise neuen und sehr vielversprechenden Zweig der Datenbanktechnologie dar. Insbesondere für hochkomplexe Anwendungsdomänen (zu denen die Verarbeitung geisteswissenschaftlicher Daten zweifellos gehört) mit auf inhaltlicher Ebene eng verwobenen Datensätzen und Quellen gelten Graphdatenbanken als geeignete Speicher- und Abfragemöglichkeit. Deswegen besteht die grundlegende Annahme, dass Graphdatenbanken gewinnbringend im Bereich der e-Humanities angewendet werden können. Bezogen auf die Arbeit mit Textkollektionen ergeben sich nun eine Reihe konkreter Forschungsfragen:

Können alternative Formen der Text- und Metadaten-Repräsentationen helfen, den Forschungsprozess flexibler zu gestalten? Wie bedingen sich digitale Repräsentationsform und Analysemöglichkeiten? Verbessern feingliedrigere Modelle die Abfragbarkeit und Interpretierbarkeit von Daten? Welches Maß an zusätzlicher Komplexität bringt die alternative Repräsentation mit sich?

Graphdatenbanken werden erfolgreich als Backend für komplexe hochskalierende Websysteme verwendet. Kann die textorientierte Forschung von dieser Technologie profitieren? Kann der parallele Zugriff auf große Textkollektionen über Graphdatenbanken realisiert werden? Können interaktive und reaktive Webportale auf Grundlage solcher Zugriffsverfahren erstellt werden?

Sind Graphdatenbanken eine geeignete Technologie für die Entwicklung von Verfahren für das Zusammenspiel quantitativer und qualitativer Betrachtungsweisen? Welche Möglichkeiten für explorative Datenanalyse bieten sich? Wie können die Verfahren verständlich kommuniziert werden? Wie kann die Dokumentation des Forschungsprozesses unterstützt werden?

Welche Aspekte der automatischen Sprachverarbeitung können durch die Textrepräsentation in Graphdatenbanken abgebildet werden? Sind *Text-Mining*-Verfahren weiterhin anwendbar? Welche Chancen und Grenzen ergeben sich für das *Information Retrieval*?

Welche über Textdaten hinausgehenden Modellerweiterungen ermöglicht die Nutzung von Graphdatenbanken? Welche Domänen und Arten von Datensammlungen lassen sich abbilden? Welche erweiterten Modelle eignen sich für graphbasierte Netzwerkanalysen? Welche zusätzlichen Aufwände entstehen im Rahmen der Anpassungen?

1.3 Beiträge zum Forschungsfeld

Für die zielgerichtete Weiterentwicklung der Digitalen Geisteswissenschaften als Anwendungsdomäne der Informatik ist die Identifizierung bisheriger technologischer Hürden und Unzulänglichkeiten eine wichtige Hilfestellung. Aus interdisziplinärer Sicht leistet das Buch darüber hinaus einen Beitrag zur Klärung der drängenden Frage, welche Art von neuartigen Werkzeugen sich aus aktuellen technologischen Entwicklungen ableiten lassen. Dadurch werden Impulse gegeben, einige der bislang durch konventionelle Analysemethoden bedingten Limitierungen der Forschungsmöglichkeiten für (textbasierte) Untersuchungsgegenstände durch alternative Ansätze zu überwinden. Die Ausarbeitung ergänzt die Demonstration der Leistungsfähigkeit neuer Methoden bewusst auch um Hinweise zu den jeweiligen Nachteilen und gibt schließlich Anregungen für interdisziplinäre Arbeitsmodi, mit denen eine projektbezogene und zielgerichtete Auswahl geeigneter Technologien stattfinden kann.

Als Teil der in diesem Buch vorgestellten Arbeiten wird ein flexibles und erweiterbares Datenmodell vorgestellt, welches geeignet ist, Texte sowie die damit verknüpften Zusatzinformationen und Analyseobjekte auf sehr feingranularer Ebene abzubilden. Es wird gezeigt, wie dieses hinsichtlich unterschiedlicher Anforderungen an die Datenabfrage

und -analyse angepasst werden kann. Dabei wird ausdrücklich nicht die Standardisierung einer einzelnen Repräsentationsform angestrebt, sondern vielmehr ein „Baukasten“ geschaffen, mit dem sich ein Pluralismus verwandter (und grundsätzlich interoperabler) Datenmodelle anhand konkreter Anforderungen herausentwickeln kann.

Auf dieser Basis wird die enge Verzahnung von digitaler Repräsentation mit der Operationalisierung digitaler Arbeitsschritte, wie etwa Persistierung, Normalisierung, Suche und Aggregation sowie mit den forschungsgeleiteten Auswertungs- und Recherche-Verfahren verdeutlicht. Die Arbeiten legen damit den Grundstein für neue Verfahren, bei denen ein stärkerer Fokus auf der Nutzung von Verknüpfungen innerhalb der Datenbasis liegt. Diese neue, „vernetzte“ Sichtweise bezieht viele unterschiedliche Aspekte in die Analysen ein und betrachtet Analyseeinheiten, wie etwa Wörter mit ihren lokalen Kontexten, ergänzende nutzerspezifische Annotationen, automatisch identifizierbare Entitäten und sprachgrenzenüberschreitende Vokabulareinheiten. Dies sind wichtige Bausteine für eine Weiterentwicklung von sprachverarbeitenden Methoden innerhalb der angewandten Informatik, insbesondere im *Text Mining* und *Information Retrieval*.

Für die Implementierung und Untersuchung dieser Verfahren wird eine prototypische Plattform entwickelt und detailliert beschrieben. Diese Plattform ist ein wichtiges Hilfsmittel, um neuartige Herangehensweisen an die Analyse von Textdaten nicht nur auf theoretischer Ebene zu diskutieren, sondern in Form nutzbarer Software in eine disziplinenübergreifende Debatte einzubringen. Sie ermöglicht die Erprobung solcher Verfahren anhand konkreter Textkollektionen und im Kontext tatsächlich existierender Forschungsfragen. Dabei besteht der Anspruch, mit dem System ein ergänzendes Werkzeug zu schaffen, das komplementär zur bestehenden Technologie eingesetzt werden kann und eine explorative Beschäftigung mit dem Datenbestand, seiner Struktur und alternativen Zugangsformen für die Analyse ermöglicht. Es soll nicht als ein Ersatz für vorhandene Werkzeuge dienen.

Darüber hinaus wird durch die Vorstellung konkreter Fallbeispiele ein Eindruck von der breiten Anwendbarkeit der entwickelten Technologie vermittelt. Anhand diverser Fragestellungen werden einerseits Querschnittsaspekte, wie der Umgang mit Metadaten und benannten Entitäten thematisiert, deren gute Verallgemeinerbarkeit auf andere Projekte offensichtlich ist. Andererseits werden auch tiefergehende spezifische Anforderungen einzelner Projekte beleuchtet und dafür technologisch kohärente Lösungswege vorgestellt, um das große Potential der Technologie für Spezialentwicklungen aufzuzeigen. Es wird gezeigt, dass Graphdatenbanken eine Bearbeitung komplexer Problemstellungen aus

den Geistes- und Sozialwissenschaften erlauben, ohne dass es zu technologischen und konzeptionellen „Medienbrüchen“ kommt.

Für die Informatik selbst ergeben sich aus den vorgestellten Arbeiten unter anderem die folgenden Impulse: Zunächst ist zu erkennen, dass die Konzeption von Textdatenmodellen stärker als bisher in die informatische Disziplin getragen werden sollte. Während aus den digitalen Geisteswissenschaften umfangreiche Vorarbeiten zum „Wesen“ von Text, seinen verschiedenen konkurrierenden Lesarten, der Anfertigung werksgetreuer Editionen sowie einer menschen- wie maschinenlesbaren Abbildung dieser Aspekte vorliegen, sind die Aspekte einer effizienten Auswertbarkeit von (großen und sehr großen) Textkollektionen auf feingranularer Ebene bisher wenig berücksichtigt. Hier besitzt die Informatik mit profunder Kenntnis von Algorithmik, Komplexitätslehre, Textstatistik und einer etablierten Praxis des Software Engineering viele Kompetenzen, die bislang zu wenig genutzt werden.

Zum anderen wird deutlich, dass die paritätische interdisziplinäre Arbeitsweise, die sich etwa in der Wirtschafts-, Bio- und anderen Formen erfolgreicher „Bindestrich-Informatik“ bereits etabliert hat, im Bereich von sozial- und geisteswissenschaftlichen Anwendungen noch gefunden werden muss. Ein Beitrag zu diesem Prozess wird mit einer technologisch transparenten, prototypenzentrischen Herangehensweise an die Entwicklung von Datenmodellen und Werkzeugen geleistet, deren digitale Artefakte über die gesamte gemeinsame Projektarbeitszeit als Kommunikationsmittel zwischen den beteiligten Disziplinen dienen können.

Der Nutzen für die Informatik aus der Beschäftigung mit den e-Humanities ist vielfältig: Neue Anwendungsgebiete mit neuen Problemen beleuchten immer auch Bereiche der Kerndisziplin, in denen weitere innovative Verfahren entwickelt werden müssen. Gleichzeitig sorgen konkrete Anwendungsfälle dafür, dass neue Verfahren aus der theoretischen wie anwendungsbezogenen Forschung in einer Weise implementiert werden, die ihre breitere Nutzung, Optimierung, Erweiterung und Dissemination fördert. Angesichts explodierender Datenmengen unterschiedlichster Strukturierungsgrade sind davon zahlreiche Teildisziplinen der Informatik berührt. Die e-Humanities mit ihrer stark quellenorientierten Arbeitsweise verlangen nach gut dokumentierten (im besten Fall selbstdokumentierenden) Verfahren und Prozessketten, die eine Provenienz für Daten sicherstellt und eine reproduktionsbereite Beschreibung relevanter Teilschritte enthält. Es kommt bei neuen Verfahren in diesem Umfeld also keinesfalls nur auf die Ausgabe, sondern auf den kompletten Prozess an.

1.4 Aufbau des Buches

An die bis hierhin vorgestellten einleitenden Überlegungen schließt sich mit [Kapitel 2](#) eine Einordnung der Arbeiten in den interdisziplinären Forschungskontext der e-Humanities an. Darin werden relevante Begriffe eingeführt und theoretisch verortet sowie die Besonderheiten der Anwendungsdomäne analysiert. Weiterhin wird der konzeptionelle und technologische Rahmen abgesteckt und ein Überblick über verwandte Felder und Ansätze gegeben.

Unter Berücksichtigung der dabei entwickelten Anforderungen und Ziele für eine methodische und technologische Weiterentwicklung von Rechercheanwendungen wird in [Kapitel 3](#) ein graphbasiertes Datenmodell für die flexible Repräsentation von forschungsrelevanten Textkollektionen konzipiert und im Kontext üblicher Text-Mining- und Information-Retrieval-Aufgaben positioniert. Darüber hinaus wird die Architektur und Funktionsweise des Recherche-Systems „Kadmos“ vorgestellt, welches durch die Nutzung dieses Datenmodells eine Vielzahl flexibler Abfrageszenarien sowie die Auslieferung vorberechnungsfreier Textstatistiken erlaubt.

In [Kapitel 4](#) werden erweiterte Anwendungsfälle für den Einsatz von Graphdatenbanken zur Unterstützung digitaler geistes- und sozialwissenschaftlicher Forschung vorgestellt, womit gezeigt wird, dass Ihre Anwendbarkeit auch deutlich über die Mindestanforderungen an forschungszentrierte Textrepräsentation hinausgeht.

Schließlich folgt im [letzten Kapitel](#) eine Zusammenfassung und kritische Betrachtung der in der Arbeit geleisteten technologischen und konzeptionellen Beiträge. Daran schließt sich ein Ausblick zu möglichen Erweiterungen und zur Einbindung in Forschungsprojekte und -infrastrukturen an.

Ein Überblick über die für dieses Buch relevanten Sammlungen von Textquellen (so genannte KORPORA) wird in [Tabelle A](#) im Anhang gegeben.

Kapitel 2

Forschungskontext und relevante Technologien

[...] think of the relationship between computing technology and the disciplines of the humanities as a moment by moment becoming of their futures, to some degree unpredictably, through an ongoing contest between the disciplines' strong sense of themselves on the one hand and all that contingently affects them, including the relentless development of digital technologies, on the other.

Willard McCarty

Professor of Humanities Computing, King's College London

aus dem Essay

„The Future of Digital Humanities Is a Matter of Words“ [McC13b]

2.1 e-Humanities

2.1.1 Entwicklung und Selbstverständnis des Fachgebiets

Die Anfänge digitaler Methoden in den Geisteswissenschaften lassen sich recht präzise zurückverfolgen und sollen hier nur kurz entsprechend [Hoc04] wiedergegeben werden. Einen detaillierteren Überblick über die Arbeiten der 1960er Jahre (aus US-amerikanischer Perspektive) bietet z. B. [Hin13].

Als Ausgangspunkt der zunächst unter dem Namen „*Humanities Computing*“ bekannt gewordenen Aktivitäten wird sehr oft das Vorhaben von Roberto Busa genannt, eine Konkordanz¹ zu den Schriften des Thomas von Aquin (und ausgewählter Texte ähnlicher Autoren) anzulegen. Da jener ein sehr umfangreiches Werk hinterlassen hat, wäre eine manuelle Bearbeitung dieses Problems sehr aufwändig: Für die Konkordanz müssen rund 11 Millionen laufende Wortformen nach ihrer Oberflächenform gruppiert und innerhalb der lexikalisch sortierten Gruppen anschließend mitsamt ihrer Positionsangaben und Umgebungswörter abgetragen werden.

Für seinen *Index Thomisticus* entschied sich Busa im Jahr 1949 deshalb, die Kooperation mit universitären Rechenzentren in den USA zu suchen, um die Arbeiten mit Unterstützung von Großrechneranlagen zu vereinfachen. Allerdings wurde sein Vorhaben von diesen Institutionen durchweg als nicht durchführbar verworfen. Schließlich fand Busa im Bereich der kommerziellen Rechentechnik in Thomas J. Watson (dem damaligen Geschäftsführer von IBM) einen großzügigen Unterstützer, der ihm die benötigte Technik und entsprechend geschultes Personal zur Verfügung stellen konnte.²

Die Arbeiten im *Humanities Computing* begannen also bereits in der frühen Anfangszeit des industriellen Großrechners, als die weithin verfügbare Technologie den Anforderungen an die Verarbeitung einer solch großen Datenmenge im Grunde noch nicht gewachsen war. Eine „Datenbank“ des *Index Thomisticus* in der damals üblichen Speicherform auf Lochkarten hätte laut Busa [Bus04] ca. 500 Tonnen gewogen.³ Das ganze Projekt war daher sehr stark auf technische Neuentwicklungen angewiesen, welche sich im Laufe der

¹ein Verzeichnis von in Texten vorkommenden Wörtern (*index verborum*), meist unter Angabe der jeweiligen Wortumgebungen (Satz oder Ausschnitt mit konstanter Anzahl an Vorgänger- und Nachfolgewörtern)

²Details zu dieser Kooperation können in [Bus80] nachgelesen werden.

³Aus den dort beigefügten hypothetischen Größenangaben ergibt sich allerdings ein reines Papiergewicht von maximal 90 metrischen Tonnen.

Zeit auch nach und nach einstellten und welche jeweils schnell adaptiert wurden – dazu Busa z. B.: „*In His mercy, around 1955, God led men to invent magnetic tapes.*“. Im Jahr 1980 konnte das Gesamtwerk schließlich (digital, im automatischen Offset-Verfahren) gedruckt werden. Es entstanden 56 Konkordanzbände mit insgesamt 65000 Seiten und gegen Ende der 1980er Jahre wurde begonnen, den *Index Thomisticus* in Form einer einzelnen Daten-CD digital zur Verfügung zu stellen.

Über Jahrzehnte hinweg wurden so enorme Aufwände in Kauf genommen, um eine wichtige Textquelle zu erschließen und in ihrer Gesamtheit besser verständlich zu machen. Dabei stellt das ursprüngliche Ziel, die Schaffung einer Konkordanz, aus heutiger Sicht nur eine sehr simple Arbeitshilfe für den Zugang zu Vokabular und seiner Nutzung in Texten dar. Durch die Digitalisierung der Texte lassen sich heutzutage alle erdenklichen Verfahren auf diese Version der Quelle anwenden. Während anfangs also ein spezifisches Werkzeug geplant war, birgt die digitale Edition das Potential, zum universellen Basiswerkzeug zu werden. Wie eingangs in dieser Arbeit beschrieben, sind aktuell unzählige Digitalisierungsinitiativen damit beschäftigt, die Grundlage für digitale Forschung in einer großen Bandbreite an Forschungsfeldern zu legen. Die Möglichkeiten der späteren Nutzung und Auswertung dieser Daten sind zum Teil noch nicht vollständig abzusehen, jedoch kann, wie gezeigt, bereits von sehr einfachen Werkzeugen großer Nutzen ausgehen. Die fachwissenschaftlichen Quellen in ihrer digitalen Form geben daher quer durch die Disziplinen neue Impulse für den Forschungsprozess.

Diese Entwicklung reiht sich in eine lange Historie von Wendepunkten geisteswissenschaftlicher Forschung ein, die unter markanten Schlagwörtern, wie dem *Linguistic Turn* (ab ca. 1900) oder dem *Spatial Turn* (ab ca. 1980) bekannt geworden sind, vgl. [BM14]. Scharloth, Eugster und Bubenhöfer verwenden in [SEB13] den Begriff „*Data-driven Turn*“ zur Charakterisierung neuer Forschungsansätze, die „auf vorgängige Hypothesen verzichten und mit Datenmengen arbeiten, die so umfangreich sind, dass sie von keinem wissenschaftlichen Individuum mehr in eine Gesamtschau gebracht werden können“. Unter Berücksichtigung der im Einleitungskapitel genannten Transformationsprozesse kann derzeit sicher ganz allgemein vom *Digital Turn* gesprochen werden.

Die von dieser Entwicklung berührten Disziplinen leisten nun unter dem Dach der *Digital Humanities* Beiträge für die Entwicklung einer gemeinsamen digitalen Methodensammlung. Dabei wird sowohl die Erprobung neuer Methodik für die Beantwortung bekannter Fragestellungen als auch das Finden neuer Forschungsfragen durch den Einsatz datenbasierter Herangehensweisen vorangetrieben. Bestehende Theorien können dadurch

kritisch hinterfragt, bestätigt oder erweitert werden und alternative Interpretationsansätze können sich herausbilden.

Der Dachverband für Aktivitäten der digitalen Geistes- und Sozialwissenschaften ist die [Alliance of Digital Humanities Organizations \(ADHO\)](#), in welcher viele regionale Verbände organisiert sind. Die weltweit agierende Community richtet über diesen Verband eine jährliche Konferenz aus. Weitere interne wie auch überregionale Konferenzen der Mitgliedsverbände und eine große Zahl themenspezifischer Veranstaltungen – von großen Konferenzen bis zu kleinen Workshops, Kolloquien und Vortragsreihen – erweitern das Angebot an physischen Begegnungs- und Austauschmöglichkeiten.

Was sind nun die Kernfragen und das Kernanliegen der Digital Humanities? Zur Diskussion der Ziele, Probleme und Ausrichtung dieser Disziplin sind bereits unzählige Einführungen, Leitartikel, Selbstbeschreibungen, Kompilationen, Gegenentwürfe und Manifeste erschienen, so beispielsweise [SSU04], [Kir10], [BDL⁺12], [Sve12], [Gol12], [AB14] und [Kön16] sowie viele weitere im Verlauf dieses Buches referenzierte Beiträge. Dennoch fällt es sehr schwer, die Charakteristika der digitalen Geistes- und Sozialwissenschaften abschließend zu definieren. Im Jahr 2012 wurden anlässlich des „DH-Tages“ die Teilnehmer gefragt: „How do you define digital humanities?“. Anhand der 256 Antworten, die daraufhin gesammelt wurden⁴ lässt sich ableiten, dass es auf der einen Seite eine Vielzahl (fachbezogener wie fachübergreifender) Hoffnungen und Wünsche an diese Forschungsrichtung gibt und allgemeine, wie auch sehr fachspezifische Sichtweisen vorherrschen – auf der anderen Seite aber auch oft gar kein Bedürfnis nach einer abschließenden Definition besteht (Antwort: „I don’t.“).

Letztlich bleiben die *Digital Humanities* daher ein Sammelbegriff für diverse Arbeiten und Ansätze, die durch meist interdisziplinär aufgestellte und mehr oder weniger stark institutionalisierte Verbünde von Forschern repräsentiert werden. Für den fachlichen Austausch haben sich zahlreiche (meist digital erscheinende) Journals⁵ und Online-Community-Systeme, wie z. B. die „DH Commons“⁶ oder der „Global Outlook DH“⁷ konstituiert. Auf die parallel dazu geschaffenen (technischen sowie organisatorischen) Forschungsinfrastrukturen wird noch in Abschnitt 2.1.3 auf Seite 24 genauer eingegangen.

⁴<http://archive.artsrn.ualberta.ca/Day-of-DH-2012/dh/>

⁵für eine Liste ohne Anspruch auf Vollständigkeit s. z. B. hier:

<http://digitalhumanities.berkeley.edu/resources/digital-humanities-journals>

⁶<http://dhcommons.org/>

⁷<http://www.globaloutlookdh.org/> – eine Initiative, die sich speziell für den Abbau von Kommunikations- und Kollaborationsbarrieren für Forscher und Forscherinnen aus Ländern mit unterschiedlichen Einkommensniveaus einsetzt

Die Charakterisierung all dieser Institutionen und Aktivitäten als „interdisziplinär“ wird stellenweise wegen der starken Abhängigkeiten und heterogenen Forschergemeinschaft sowie häufigen Mischqualifikationen der einzelnen Forscher sogar durch das Prädikat „transdisziplinär“ ersetzt. So beschreibt Lin etwa in [Lin12] unter Bezugnahme auf die Untersuchungen zur Wissensproduktion in [GLN⁺94] eine sich neu herausbildende Arbeitsweise, „*which is context-driven, problem-focused, and transdisciplinary, involves multidisciplinary teams with heterogeneous backgrounds working together. This differs from traditional [...] research that is academic, investigator-initiated and discipline-based knowledge production.*“.

Die Wahrnehmung der *Digital Humanities* als aufstrebende Fachdisziplin oder neue wissenschaftliche Strömung stößt im heutigen akademischen Umfeld allerdings auf eine sehr gemischte Rezeption:

[...] it bloomed into an umbrella for a range of work taking place across the humanities, an intellectual turn towards both exploiting and understanding computational and networked technology, and – depending on who you ask – a category of research that either holds in its hands the future of the humanities or is complicit in a neo-liberal agenda intent on destroying higher education and all that humanists hold dear. ⁸

Der Großteil der Kritik zielt dabei vorrangig auf Aspekte der übermäßigen Institutionalisierung und der damit verbundenen Verschiebung von Förderprioritäten ab, einige Beiträge beziehen darin jedoch explizit auch die gesamte Praxis und Community der *Digital Humanities* ein, wie etwa [ABG16].

Ungeachtet dieser Punkte kann die offene fachübergreifende Kommunikation und Kollaboration, wie sie in den *Digital Humanities* praktiziert wird, als wichtiges, fruchtbares und zukunftssträchtiges Forschungsmodell angesehen werden. Bei der Suche nach adäquaten mathematischen, technologischen und informationswissenschaftlichen Rahmenbedingungen für künftige Arbeiten eröffnet sich insbesondere für die Informatik ein vielschichtiges, komplexes, datenreiches und innovationsbereites Anwendungsfeld. Das stetig wachsende Interesse innerhalb der Informatik verdeutlicht sich u. a. auch durch entsprechende Schwerpunktsetzungen für etablierte Konferenzen und Publikationsreihen. In [HHSH15] wird als einleitender Beitrag zum Problemkreis des „Informationsmanagement für *Digital Humanities*“ betont, dass dabei als zwei untrennbare Aspekte die „[...]

⁸aus einem Buch-Review von James Baker: <http://www.history.ac.uk/reviews/review/1634>

Unterstützung geisteswissenschaftlicher Forschungsarbeiten mit Methoden der digitalen Informationsverarbeitung sowie Forschungsfragen, die sich dadurch auch für die Informatik ergeben“, relevant sind.

Einen Schritt weiter gehen die *Computational Humanities*, in deren Kern die Entwicklung neuer Algorithmen, Formalismen und generischer Methoden zur Verarbeitung von Quellenmaterial steht. Hier ist die konkrete Anwendung der erstellten Werkzeuge in den Geistes- und Sozialwissenschaften zum Teil erst ein nachgelagerter Schritt einer eigentlichen informatiknahen Grundlagenforschung.

Aus dieser Vielfalt an Perspektiven ergibt sich auch die Verwendung des (sonst eher seltener genutzten) Begriffs *e-Humanities* im Kontext dieses Buches. Wie in Abbildung 2.1 verdeutlicht, wird damit eine beide „Pole“ berücksichtigende Gesamtbetrachtung und Mediation zwischen den aus historischen und methodischen Gründen zum Teil weit entfernten Forschungsparadigmen beabsichtigt.

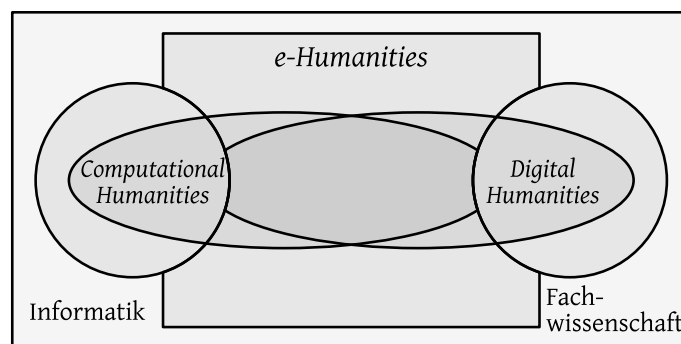


Abbildung 2.1: Einordnung der *e-Humanities* ins Fächergefüge, sinngemäß übernommen aus [HI14]

Ein weiterer positiver Aspekt dieser Bezeichnung ist die direkte begriffliche Assoziation zu den *e-Sciences*. Diese stehen für eben jenes neue und universelle Forschungsverständnis, das eine offene Kollaboration von Fachwissenschaftlern über digitale Infrastrukturen in den Mittelpunkt stellt. Nicht zuletzt kann auch die dort propagierte Reproduzierbarkeit von Analysen⁹ als wünschenswerter Input für digitale geisteswissenschaftliche Forschung angesehen werden.

⁹vgl. z. B. die zahlreichen Anwendungsfälle in [FFR16]

2.1.2 Forschungsmethodik und aktuelle Entwicklungen

Die aktuellen Diskussionen und neuen Methoden in den e-Humanities stehen stark unter dem Einfluss der von Franco Moretti geprägten Einteilung von analysierenden und interpretierenden Forschungsaktivitäten in ein *Close Reading* und ein *Distant Reading*, vgl. [Mor13]. Damit wird ein Kontinuum zwischen qualitativen und quantitativen Verfahren sowie deren Anwendung auf große, allgemeine aber auch kleine und spezialisierte Quellensammlungen sowie zwischen automatischen und manuellen Analyseprozessen aufgespannt. Projekte, die neue Ansätze zur Adaption digitaler Methodik entwickeln, positionieren sich oft in feststehende Nischen innerhalb dieses Raums, um damit ihre generelle Ausrichtung und ggf. Anknüpfung an bestehende Arbeitsweisen zu kommunizieren.

Die literaturwissenschaftlich geprägte Richtungsvorgabe Morettis sollte darüber hinaus jedoch vielmehr als Anregung für eine flexible und bedarfsgerechte Methodensuche und Methodenkombination sowie die Definition von *Best Practices* innerhalb des sich neu ergebenden Spektrums digitaler Ansätze und Werkzeuge dienen. Stulpe und Lemke schlagen in diesem Sinne für die Sozialwissenschaften (in welchen Texte selbst nur als sekundäre Forschungsgegenstände gelten können) unter der Bezeichnung „*Blended Reading*“ eine Verbindung (semi-)automatischer Verfahren und manueller Detailanalyse vor, welche auf die Nutzung tatsächlich vorhandener „Erkenntnispotenziale“ abzielt, s. [SL16].

Entsprechende breitere Ansätze werden auch in der Literaturwissenschaft selbst diskutiert, etwa unter dem Stichwort der „*Macroanalysis*“ bei [Joc13] oder mit den in [Ram11] beschriebenen Ansätzen – „*toward an algorithmic criticism*“. Während die theoretischen Überlegungen dort sehr weitreichend und aus Sicht der Fachwissenschaft sehr anregend sind, bewegen sich der Grad der Formalisierung und die algorithmischen Beiträge (anders als in den Sozialwissenschaften) dort eher auf niedrigem Niveau – etwa bei einfacher Wortstatistik. Auch in seinem (aus Sicht der Informatik) zunächst sehr technisch klingenden Buch „*Graphs, Maps, Trees: Abstract Models for a Literary History*“ [Mor05] führt Moretti keine eigentlichen Formalismen im Sinne der „exakten“ Wissenschaften ein. Es werden unter diesem Titel lediglich Kartierung und das Anfertigen von Diagrammen als Arbeitstechnik des *Distant Reading* charakterisiert¹⁰. Allgemein ist in einer Vielzahl

¹⁰Mit dem Terminus „*Graphs*“ werden in Morettis Buch Diagramme und Schaubilder bezeichnet, nicht Graphen im Sinne dieses Buches.

von Arbeiten jedoch allmählich auch ein Trend zur Adaption komplexerer Verfahren zu erkennen.

In jedem Fall ist ein wesentlicher Aspekt für die Etablierung digitaler Forschungsansätze die damit erreichbare methodische Transparenz. Eine in diesem Umfeld häufig genutzte Begrifflichkeit ist die der „Black Boxes“, die informell in etwa so beschrieben werden können: Werkzeuge, die für die Bearbeitung einer bestimmten Fragestellung herangezogen werden, wobei deren interne Wirkmechanismen dem Nutzer nicht oder nur ungenügend bekannt sind.

Dabei handelt es sich grundsätzlich um abgeschlossene Systeme, deren Verhalten zwar untersucht und beschrieben werden kann, deren Wirkweise jedoch unklar ist. Die Ursprünge dieser Betrachtungsweise liegen in der Systemtheorie und Kybernetik. Im gleichnamigen Kapitel in [Ash56] wird die *Black Box* als ein experimenteller Untersuchungsgegenstand vorgestellt, der mit verschiedenem *Input* konfrontiert werden kann und dessen *Output* sich messen oder beobachten und in Bezug zum Input protokollieren lässt. Daraus können Hypothesen über die Wirkweise (und somit indirekt über den inneren Aufbau) entwickelt werden.

In den e-Humanities ist der eigentliche Untersuchungsgegenstand jedoch nicht die *Black Box* selbst, sondern deren Input (oder die Objekte, aus denen sich der Werkzeug-Input ableitet). Die Unkenntnis der *Black Box* macht Ergebnisinterpretation in diesen Fällen schwierig: Wenn nicht klar ist, welche Aspekte der Eingaben wie verarbeitet werden, um eine Ausgabe zu generieren, kann das Werkzeug nicht sinnstiftend verwendet werden. Eine methodische Transparenz erfordert insbesondere auch eine Transparenz der verwendeten Datenrepräsentation und Datentransformation. McCarty schreibt dazu in [McC13a]

„Thus digital representation does not matter to the person interested only in output or effects. But it is crucial to the person, [...] who wants to know what is lost in translation, and more importantly what that loss illumines.“

Daraus lässt sich ableiten, dass nicht nur ein Verständnis der grundsätzlichen Funktionsweise und Ausgabe eines Verfahrens notwendig ist, sondern auch die Art und Weise der Ableitung von Informationen über den Aussagegegenstand, dessen Diskretisierbarkeit und Formalisierbarkeit bedacht werden muss. Die Repräsentation muss relevante Aspekte abbilden können und in einer Form vorhalten, die verständliche, nachvollziehbare Analysen ermöglicht.

Diese Sichtweise reiht sich in aktuelle Bestrebungen ein, Arbeitsabläufe und Methoden in den e-Humanities systematisiert zu beschreiben und ihr Zusammenspiel mit verschiedenen Formen von Datenquellen zu charakterisieren. Beispielsweise wird aktuell mit der „[Taxonomy of Digital Research Activities in the Humanities \(TaDiRAH\)](#)“¹¹ ein Versuch der (mehrsprachigen) Definition von typischen geisteswissenschaftlichen Forschungsaktivitäten unternommen, s. u. a. [BDPS16]. Solche Bestrebungen zur Ordnung und stückweisen Formalisierung zentraler Arbeitsabläufe sind seit der Jahrtausendwende im Gespräch, vgl. etwa [Uns00]. Eine direkte Umsetzung in Interaktionsmuster mit digitalen Werkzeugen existiert jedoch bisher nicht, zumal noch nicht abschließend geklärt ist, wie in Forschungsprojekten üblicherweise das Verhältnis von (gut verallgemeinerbaren) Kernabläufen zu (thematisch und methodisch sehr diversen) vorhabensspezifischen Arbeitsschritten gewichtet ist. Generell trägt die explizite Beschreibung aber zur individuellen Projektplanung auch im Hinblick auf die zu wählende Repräsentationsform bei.

Die Entwicklung von Vorgehensmodellen kann die methodische Transparenz befördern, wenn es dabei gelingt, *Best Practices* aus der generischen Methodenentwicklung und vorhergehenden Projekterfahrungen mit einer zielstrebigem, stark von den Forschungsfragen geleiteten Herangehensweise zu kombinieren. Als ein erster Schritt hin zu solchen „Projekt-Blaupausen“ kann beispielsweise die in [Fec16] vorgestellte „*Data Adaption*“ angesehen werden, welche ein mehrschichtiges Vorgehensmodell, bestehend aus Phasen der Beschäftigung mit den folgenden Aspekten vorsieht:

- Forschungsziel und Quellen
- Beschreibungsmodell
- Datenmodell
- Datensammlung und Anpassung
- Exploration
- Detailanalyse

Die einzelnen Schritte bauen jeweils auf den in der vorhergehenden Ebene gewonnenen Erkenntnissen und getroffenen Entscheidungen auf. Diese sollten jeweils auch phasenweise dokumentiert werden, um in dieser Form als weitere Orientierungshilfe für die nachfolgenden Arbeiten zu dienen.

Da in diesem Buch ein generisches Werkzeug beschrieben wird, lässt sich der Fokus nicht

¹¹<http://tadirah.dariah.eu/vocab/index.php>

auf die ersten beiden (projektspezifischen) Schritte legen. Die Beiträge setzen daher als erstes bei der Entwicklung eines flexiblen Datenmodells an. Der Begriff des Datenmodells wird dabei in Anlehnung an die in [FJ15] verwendeten Definitionen wie folgt verstanden: Erstens als Ausdruck einer konzeptuellen und logischen Sicht auf die zu erforschenden Daten, zweitens als Kommunikations- und Konkretisierungs-Werkzeug für die datengetriebene fachliche Bearbeitung der Fragestellung und drittens als technische Beschreibungsform für die (inhärente oder angenommene) Struktur von Daten. Für letzteren Aspekt positionieren Flanders und Jannidis das Datenmodell zwischen Modellinstanzen, also die konkreten im Modell gespeicherten Daten, und ein Metamodell, also die „Modellierungssprache“ mit ihren als Bausteine dienenden Modellierungskonstrukten. Zurecht bemerken sie den oft sehr unterschiedlich ausgeprägten Grad der Nutzung der im Datenmodell definierten Konstrukte in einzelnen Instanzdatensätzen. Nicht immer sind Datensammlungen tatsächlich so komplex wie das ihnen zu Grunde liegende Schema.

Am Ende der Projektentwicklung in den e-Humanities steht üblicherweise die Detailanalyse. Oft ist zumindest zu Beginn dieses Stadiums der thematische Fokus und der zu betrachtende Ausschnitt der Quellen noch nicht so genau eingegrenzt, dass eine klare Forschungsagenda für die schrittweise Abarbeitung von Einzelbelegen nach einem festen Schema erstellt werden könnte. Abweichend zu [Fec16] soll hier die Exploration der Quellen bis zuletzt als Bestandteil des Forschungsprozesses angesehen werden.

Den Ressourcen-Zugriff ab einem bestimmten Zeitpunkt hauptsächlich auf bereits bekannte oder durch Vorsondierung stark eingegrenzte Teilbereiche der Quellensammlung zu beschränken, ist sicher konform mit traditionellen, forschungsfragengeleiteten Herangehensweisen, sollte aber durch datengetriebene Ansätze und alternative, explorative Zugänge zu den Daten ergänzt werden. Aus dem Bereich der explorativen Suche ist bekannt, dass sich damit natürlich nicht unerhebliche Entwicklungsaufwände ergeben. So heißt es etwa in [Mar06]:

Research tools critical for exploratory search success involve the creation of new interfaces that move the process beyond predictable fact retrieval.

Aus der Schaffung explorativer Zugänge erwachsen jedoch nicht selten unerwartet Erkenntnisse über Einzelinformationen oder Zusammenhänge in den Daten, die mit zielgerichteteren Verfahren (ohne Kenntnis ihrer Existenz) nicht gefunden worden wären. Dieser sogenannte Serendipitätseffekt kann eine wesentliche Rolle im Forschungs-

prozess spielen. Er ist z. B. in [BQHR12] näher beschrieben, dort im Bezug auf Online-Rechercheinstrumente.

Die Rechercheaktivität ist somit weniger als Ansammeln von Fakten, sondern als schrittweises Definieren von Kontexten anzusehen, innerhalb derer sich potentiell interessante Fakten ergeben. Zur Detektion, Modellierung und Analyse dieser Kontexte kann auf etablierte Methoden der Informatik zurückgegriffen werden. Beispielsweise haben sich statistische Methoden zur Analyse unstrukturierter Daten, vgl. [MS99], im Kontext der e-Humanities bewährt. Es werden aber verstärkt komplexere Instrumente benötigt, so dass sich aus der Anwendungsdomäne heraus auch innerhalb der Informatik disziplinenübergreifende Problemstellungen ergeben. Das betrifft neben der Sprachverarbeitung auch die Forschung an Datenbanken, Datenintegration, Datenmodellierung und Wissensrepräsentation, Informationsvisualisierung und Big Data. In diesen Schwerpunkten ergibt sich jeweils weiterer Forschungsbedarf sowie allgemein die Notwendigkeit eines Austauschs der Gebiete untereinander.

Eine weitere aktuelle und in diesem Zusammenhang wichtige Entwicklung ist die verstärkte explizite Betrachtung komplexerer Zusammenhänge im Forschungsprozess. Eng gekoppelt daran ist eine Interpretation der Quellen und Forschungsobjekte in Form von Netzwerken. Diese Herangehensweise wird später in diesem Kapitel noch ausführlicher vorgestellt. Instrumente, die einen interaktiven und visuellen Zugang zu (kleinen und mittelgroßen) Netzwerken bieten, werden in den e-Humanities immer populärer, allen voran Gephi [BHJ09]. Dadurch ist ein sehr intuitiver Umgang mit Netzwerken möglich. Dieser Umstand blendet jedoch möglicherweise aus, dass bisher nur wenige Arbeiten zu grundlegenden Aspekten der Datenmodellierung und Formalisierung für die Arbeit mit Netzwerken kultureller Artefakte und geistes- wie auch sozialwissenschaftlicher Forschungsgegenstände existieren.

2.1.3 Forschungsressourcen und -infrastrukturen

Wie bereits in der Einleitung angeklungen ist, kann Forschung im Kontext der e-Humanities nur selten als solitäre Aktivität einzelner Akteure durchgeführt werden. Größere Projekte, aber auch individuelle Forschungsaktivität, werden Teil eines allgemeineren Forschungsprozesses, der innerhalb einer Forschungscommunity stattfindet – wobei die traditionellen, nicht auf Kollaboration ausgerichteten Forschungsaktivitäten dadurch nicht ersetzt, aber um viele perspektiverweiternde Möglichkeiten und digitale Instrumente

ergänzt und damit unterstützt werden.

Aus Sicht der Wissenschaftsförderung ist diese Tendenz eine willkommene Entwicklung, wird darin doch die Möglichkeit zur Synergiebildung und Vermeidung von Mehrfachaufwendungen für gleiche Arbeiten gesehen. Gerade für die grundlegenden Digitalisierungsprozesse und die daraus resultierenden Datenbestände wird eine breite Nutzung in verschiedensten Fachkontexten angestrebt. Das umfasst insbesondere auch die gleichzeitige Nutzung von Daten aus mehreren Digitalisierungsinitiativen und Datenrepositorien. Nach den „[DFG-Praxisregeln](#)“ ist das Ziel systematischer Digitalisierung „[...] nicht nur das Bereitstellen, sondern auch und vor allem das Vernetzen der unterschiedlichen Ressourcen zu einer virtuellen Forschungsinfrastruktur.“ [\[DFG13\]](#)

Bei der Vielzahl der verfügbaren, meist verteilt erstellten Ressourcen fällt es im Forschungsprozess oft schwer, einen umfassenden Überblick über relevantes Material zu erhalten. Damit durch die Ressourcen auch tatsächlich Mehrwerte für die Forschung entstehen können, müssen sie zuallererst gut auffindbar vorgehalten werden. Verschlagwortung, Kategorisierung und die Erstellung von Beschreibungstexten bilden dabei wichtige Zugangsmechanismen. Darüber hinaus wird es auch immer wichtiger, inhaltliche Verbindungen zwischen Ressourcen und zu Forschungsgegenständen (Personen, Institutionen, Epochen, Genres, etc.) nachvollziehbar und für das Auffinden der Ressource nutzbar zu machen.

Um diesen Anforderungen gerecht werden zu können, ist beim Umgang mit Forschungsressourcen und insbesondere bei ihrer Bereitstellung innerhalb von Forschungsinfrastrukturen die Verwaltung von Metadaten eine wesentliche Schlüsselaufgabe. Dabei gilt es, eine große Bandbreite an Aspekten zu erfassen, die irgendwann im Recherche-, Auswertungs- oder Interpretationsprozess relevant werden könnten. Neben selbstverständlichen Angaben, wie dem Autor eines Werks, der Sprache eines Textes oder dem Material eines Objekts, kommen je nach Anwendungsfall viele weitere Informationen in Betracht, wie die wissenschaftliche Datierung, frühere Katalogisierungsnummern, Provenienzinformationen und ggf. Erwerbungs historie, Angaben zur Dokumentation des Digitalisierungsvorgangs, Lizenzinformationen für die Verbreitung usw.

Als eine erste Erschließungsstufe auf dem Weg zur Forschungsinfrastruktur können in dieser Beziehung Portale für die Vernetzung von Sammlungen kulturellen Erbes (wie z. B. Gemälde, Texte, museale Objekte und Fotos) angesehen werden, beispielsweise das

europäische Projekt Europeana¹². Mit der Deutschen Digitalen Bibliothek¹³ existiert ein vergleichbares deutschlandweites Projekt mit einer Schwerpunktsetzung auf textuelle Quellen. Bei diesen Portalen handelt es sich um einen virtuellen Zusammenschluss der Bestände vieler einzelner Institutionen. Damit dies ohne ausufernde Zusammenführungsaufwände vonstatten gehen kann, wird für diese (verteilten) Systeme ein gemeinsames konzeptionelles Modell benötigt.

Für diesen Zweck hat sich das (recht allgemein gehaltene) [Conceptual Reference Model \(CRM\)](#) der [International Committee for Documentation of the International Council of Museums \(CIDOC\)](#) etabliert, zu welchem z. B. auch „Übersetzungen“ (sogenannte *Mappings* für das Europeana-Datenmodell) existieren. Auch für die konkreten Wertebereiche für die Beschreibung von Objekten existieren Ressourcen, etwa die Vokabulare und Thesauri des Getty Research Institute¹⁴. Für einige Eintragstypen, wie Künstlernamen, enthält diese Sammlung auch Normdaten – also Einträge mit Identifikationsnummer, Listen von Benennungsvarianten und Zusatzinformationen. Für solche Normdatenrepositorien existieren ebenfalls Portale für den Zusammenschluss über Institutionsgrenzen hinweg. Im [Virtual International Authority File \(VIAF\)](#) sind die gemeinsamen Normdatensätze vieler Bibliotheken und anderer Forschungsinstitutionen zusammengefasst und verknüpft.

Eine Forschungsinfrastruktur umfasst jedoch mehr als nur vereinheitlichte Sammlungen von Datenrepositorien und Referenzmodellen: Sie sind Ankerpunkt und Koordinationsstelle für eine heterogene Landschaft aus Quellen, Services und Konsumenten. Für die e-Humanities existieren auf europäischer Ebene in der Hauptsache zwei große Infrastruktur“-Initiativen: [Common Language Resources and Technologies Infrastructure \(CLARIN\)](#) und [Digital Research Infrastructure for the Arts and Humanities \(DARIAH\)](#). Beide ergänzen sich in ihren Schwerpunkten und stehen auf technischer und organisatorischer Ebene in regem Austausch, unterstützt auch durch die europäische Initiative zur Synergiebeförderung zwischen diesen Infrastrukturprojekten, namens [Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies \(PARTHENOS\)](#).

Einen Einblick in das Servicekonzept einer solchen Forschungsinfrastruktur gibt z. B. [\[WAB⁺09\]](#) mit der Vorstellung von (zu diesem Zeitpunkt bekannten) Anforderungen an Webservices und Methoden der Prozesssteuerung in [CLARIN](#). Damit Infrastrukturen

¹²<http://www.europeana.eu/>

¹³<http://www.deutsche-digitale-bibliothek.de/>

¹⁴<http://www.getty.edu/research/tools/vocabularies/>

Einträge aus verstreuten Ressourcen und Repositorien auffinden können, existieren Mechanismen zum *Harvesting* der relevanten Metadaten. Hierbei kommt häufig das von der [Open Archives Initiative \(OAI\)](#) entwickelte und von zahlreichen Bibliotheken und Archiven übernommene [OAI Protocol for Metadata Harvesting \(OAI-PMH\)](#) zum Einsatz. Entsprechend müssen sie mit den vielen verschiedenen Modellierungsmöglichkeiten für Metadaten und deren konkreter Semantik umgehen können – Aspekte, die später im Buch noch mehrfach zur Sprache kommen werden.

Für die textorientierten e-Humanities ergeben sich die meisten der für die Infrastrukturprojekte relevanten Ressourcen aus der großflächigen Digitalisierung von Manuskripten oder Retrodigitalisierung gedruckter Werke. Daraus entstehen allgemeine oder themenspezifische Textsammlungen, sogenannte Korpora, wie sie beispielsweise vom Deutschen Textarchiv¹⁵ in großem Umfang angeboten werden. Andere Arten von textuellen Quellen bilden digitale (wissenschaftliche) Texteditionen bekannter Werke, an welchen stets neuer Bedarf herrscht. Mit fortschreitendem Forschungsstand werden immer neue Aspekte für die Editionen wichtig und aus der z. T. komplexen Forschungsmaterie lässt sich nur selten eine eindeutige oder „optimale“ Empfehlung zur Editionspraxis ableiten. In [\[Cle15\]](#) heißt es darüber hinaus:

„[...] how a text is edited, in respect of any norms and standardisations, is a statement about the nature of the text and its tradition, transmission, and history; and such a statement may be true or false.“

Die damit verbundenen Entscheidungen und deren Implikationen sind vom Standpunkt des Betrachters und vom konkreten Anwendungsfall abhängig. In der akademischen Anwendungspraxis ist ein Pluralismus von diesbezüglichen Herangehensweisen durchaus erwünscht. Entsprechend existieren auch kleinere Infrastrukturprojekte für fachspezifische Editionsbelange (sowie Projekte zur Korpuserstellung) – s. z. B. [\[Bab11\]](#) für die „Digitalen Altertumswissenschaften“. Die dort entwickelten Ressourcen, Werkzeuge und anderen Services können perspektivisch mit größeren Initiativen zusammengeschlossen werden. In [CLARIN](#) wurden für spezifische Anwendungsfelder sogenannte Facharbeitsgruppen eingeführt, um solche Formen der Integration zu befördern. Von diesem methodischen Übertrag profitieren im Idealfall auch andere Anwendergruppen.

Aus den angesprochenen Digitalen Altertumswissenschaften stammt z. B. aus dem Umfeld der Edition klassischer griechischer Texte ein spezialisierter Service und eine darauf

¹⁵<http://www.deutschestextarchiv.de/>

aufbauende Infrastruktur namens „[Collections, Indices, Texts, and Extensions \(CITE\)](#)“, s. [SW09]. Diese definiert [Canonical Text Services \(CTS\)](#), in welcher Referenzierungen von Textstellen über Webtechnologie in Form eines eindeutigen [Uniform Resource Name \(URN\)](#) möglich ist. Dieser Ansatz wird mittlerweile auch für Anwendungen außerhalb der Altertumswissenschaften genutzt und soll perspektivisch in [CLARIN](#) integriert werden.

Aspekte, die – ganz allgemein – für die Nutzung von Textsammlungen in den e-Humanities relevant sind, sowie die dafür derzeit verfügbaren digitale Repräsentationsformen werden in den folgenden Abschnitten detaillierter vorgestellt.

2.2 Text- und Korpusrepräsentation

2.2.1 Charakterisierung von Forschungskorpora

Der in diesem Buch häufig verwendete Begriff des KORPUS für Sammlungen von Texten lehnt sich an die Verwendung großer Textsammlungen in der Korpuslinguistik an. Diese zielt auf ein Verständnis der Funktionsweise von Sprachen (und Sprache an sich) über datengeleitete Verfahren ab, wobei die Größe der genutzten Datensammlungen eine Bearbeitung mit digitalen Mitteln erforderlich macht. In der Linguistik wurden bereits seit den späten 1950ern zahlreiche Berührungspunkte mit der diskreten Mathematik und der Informatik identifiziert und genutzt, etwa im Bereich formaler Grammatiken¹⁶. Die Korpuslinguistik, die seit den 1990er Jahren stetig an Popularität gewinnt, macht sich solche Vorarbeiten jedoch nicht unbedingt zu eigen, sondern orientiert sich eher an der Statistik und an von Experten gelenkter Muster-Abfrage.

Diese Auffassung des Textbestandes als Arbeitsgrundlage für quantitative Betrachtungen der Artefakte kultureller Prozesse hat sich in den letzten Jahren verstärkt in viele Gebiete der Geisteswissenschaften übertragen. Korpora werden heutzutage in großem Maße für die historische, politologische oder sozialwissenschaftliche Forschung herangezogen. Den in diesem Buch beschriebenen Arbeiten liegt ebenfalls ein solcher erweiterter und allgemeinerer Korpusbegriff zugrunde. Er umfasst kollaborative Text-Repositoryn, digitalisierte Bucheditionen, abgeschlossene Textkollektionen aber auch dynamisch erweiterte Quellensammlungen.

¹⁶s. z. B. [Mar98]

Hauptziel beim Zusammentragen von Dokumenten und Erstellen digitaler Forschungskorpora ist das Anlegen großer Sammlungen relevanter Texte, in denen aussagekräftige statistische Untersuchungen möglich sind. Je größer ein Textbestand, umso größer die Chance zur Abdeckung des Gesuchten. Die (quantitativ und qualitativ) passende Auswahl von Quellen zur Beantwortung einer Forschungsfrage ist ein immer wieder unterschätzter Arbeitsschritt in den e-Humanities, wie schon in [Fec16] festgestellt wird. Auch hier können Lehren aus der Korpuslinguistik übernommen werden. Sehr plakativ wird dieser Umstand z. B. in [MH12] mit dem Beispiel „[...] *there would be little point in exploring the noun classification system of Swahili by looking in a corpus of English newspaper texts.*“ umschrieben.

Die Rolle von Korpora im Forschungsprozess ähnelt in den e-Humanities der im linguistischen Bereich. Köhler betrachtet diese in [Köh05] als eine Form von „Evidenzquellen“ und definiert eine Vielzahl von Anforderungen an die Stichproben-Eigenschaften linguistischer Korpora¹⁷. Die Auswahlkriterien entsprechen dem Wunsch nach einer wahrheitsgetreuen quantitativen Auswertung von Sprachphänomenen. In den meisten anderen Disziplinen werden Korpora auch als Evidenzquelle angesehen, wobei jedoch nicht in jedem Fall so strenge Anforderungen an die Stichprobeneigenschaft der Korpora gestellt werden oder gestellt werden können. In historisch arbeitenden Disziplinen wird in der Regel zu Anfang eines Forschungsvorhabens keine künstliche Verknappung des Quellenmaterials durchgeführt. Jede einzelne Quelle kann – auch wenn sie nicht als „repräsentativ“ gelten kann – Informationen enthalten, die geeignet sind, die Interpretation des Gesamtbestandes stark zu beeinflussen. Es soll mit Korpora dort also keine Form der quantitativen Forschung betrieben werden, die sich vordergründig auf die Beschreibung des Regelfalls (und damit ggf. des „Offensichtlichen“) konzentriert.

Angesichts des zum Teil extrem fragmentarischen und lückenhaften Materials, auf das für spezifische Fragestellungen zurückgegriffen werden muss, kann weder Repräsentativität noch Vollständigkeit (in irgendeinem Sinne außer im Bezug auf die Überlieferungslage) als realistisches Ziel für den Korpusaufbau angegeben werden. Die teils schlechte Quellenlage ist jedoch nur ein Aspekt der potentiellen Hemmnisse für korpusbasierte sozial- und geisteswissenschaftliche Forschung. Ebenso schwierig gestaltet sich Forschung im Angesicht unklarer Lizenzfragen oder hoher Lizenzgebühren. Offene, frei nutzbare Korpora mit wissenschaftlichem Anspruch hinsichtlich der Daten- und Metadatenkompilation sind nach wie vor selten.

¹⁷u.a. Repräsentativität, Homogenität und Normalverteiltetheit von zu untersuchenden Phänomenen

Auf der Seite der Werkzeuge kann festgestellt werden, dass sich aus einer Reihe von Spezialentwicklungen für einzelne Korpora über die Zeit generische Ansätze entwickelt haben. Solche Korpusverwaltungssysteme konnten sich bereits in den 1990er Jahren etablieren und sind bis heute im Einsatz. Auf viele Systeme kann dabei über die [Corpus Query Language \(CQL\)](#), s. z. B. [CS96], zugegriffen werden. Damit können je nach Datenbasis sowohl Texte bestehend aus mehreren Zeichenketten- und Annotationsebenen ausgewertet werden als auch die Informationen aus Abhängigkeitsgraphen in die Abfrage einbezogen werden. Als ein häufig genutztes Werkzeug dieser Kategorie ist Sketch Engine¹⁸ sowie eine eingeschränkte, aber frei nutzbare Version davon, namens NoSketch Engine¹⁹, aufzuführen. In der „XML Aware Indexing and Retrieval Architecture“²⁰ werden CQL-Anfragen in einer auf der [Extensible Markup Language \(XML\)](#) basierten Form verarbeitet und die Korpora werden im XML-Format vorgehalten.

Diese Werkzeuge konzentrieren sich in den meisten Fällen auf linguistische Aspekte der einzelnen Texte und sind daher meist auf die Ausgabe von Textstellen und damit verbundene Statistiken ausgelegt. Die explorative Erschließung von Dokumentkollektionen, wie sie in den e-Humanities im Vordergrund steht, ist damit – auch wegen unzureichender Möglichkeiten, Metadaten innerhalb der Korpora zu verwalten – nicht komfortabel möglich. Auch für Verfahren des Text Mining sind die gebotenen Schnittstellen zu den Korpusdaten oft nicht ausreichend.

Für die Arbeit mit Korpora in den e-Humanities existieren keine umfassenden generischen Werkzeuge, da sich die Anwendungsfälle und Vorgehensweisen oft stark unterscheiden. In diesem Umfeld kommen oft verschiedene Ansätze zur Formalisierung der Fragestellung zur Anwendung, welche auch unterschiedliche Anforderungen an die digitale Repräsentation der Daten und an Möglichkeiten für den Zugriff darauf stellen. Die Texte sollen nicht nur maschinenlesbar, sondern maschinenauswertbar vorgehalten werden. Das heißt, dass darauf Verfahren zur statistischen Auswertung angewendet werden sollen, die die Korpusexploration lenken können. Das kann die Empfehlung „ähnlicher“ Dokumente oder Textstellen, „ähnlicher“ Vokabeln oder „ähnlicher“ Satzstrukturen sein. Die gewünschten Ähnlichkeiten sind dabei meist projektspezifischer Natur.

Die explorative Arbeit führt üblicherweise zu interessanten Teilproblemen der Ausgangsfragestellung, welche dann isoliert betrachtet werden sollen. Dafür hat sich das Anlegen

¹⁸<http://www.sketchengine.co.uk/>

¹⁹<http://nlp.fi.muni.cz/trac/noske>

²⁰<http://sourceforge.net/projects/xaira/>

sogenannter Subkorpora etabliert. Diese sind Kopien von Teilen des Originalkorpus und beschränken sich jeweils auf fest definierte Autoren, Genres, Zeiträume oder ähnliche aus Metadatenabfragen oder manueller Auswahl gebildete Kriterien. Subkorpora können sehr klein und speziell sein. Auf der anderen Seite ergeben sie sich zuweilen auch durch das Weglassen kleiner spezieller Teile und gleichen so in Größe und Aufbau stark dem Ursprungskorpus. Es lässt sich in diesem Umfeld tatsächlich wenig Verallgemeinerndes zur Nutzungsweise von Dokumentkollektionen sagen. Textkorpora in den e-Humanities enthalten zudem verschiedenste Textgattungen, und decken Material diverser Forschungsfelder ab. Wenig einheitlich sind letztlich auch die in ihnen verwendeten Sprachen und Alphabete. In den folgenden Abschnitten sollen die in diesem Zusammenhang relevanten Aspekte der digitalen Repräsentationsformen für textuelle Quellen vorgestellt werden.

2.2.2 Zeichenrepräsentation

Geschriebene Sprache hat sich als Kulturtechnik erst allmählich etabliert. Lange Zeit wurde Sprache ausschließlich gesprochen und war damit ein direktes Kommunikationsmittel, das keinen physischen Träger besitzt und so auch keine archivierende Form der Externalisierung von Wissen darstellt. Während zu Beginn der Schriftnutzung hauptsächlich genau diese langfristig bewahrende Eigenschaft der Schrift zur Dokumentation wichtiger staatsgeschichtlicher oder religiöser Aussagen im Vordergrund stand²¹, etablierte sich nach und nach die Nutzung von Schrift in der alltäglichen Kommunikation und als Verwaltungswerkzeug – geschrieben auf leicht transportablen und massenweise verfügbaren Trägermedien.

Über viele Kulturen und Sprachen hinweg hat sich der Prozess der Verschriftlichung über die Jahrhunderte und Jahrtausende ganz unterschiedlich vollzogen und so existieren in der Folge auch sehr unterschiedliche Systeme der Nutzung von Schriftzeichen. Im Wesentlichen lassen sich zwei Schriftsysteme unterscheiden: In der Logographie weist jedes Zeichen eine gesonderte Bedeutung auf, die sich mit der anderer Zeichen zu komplexeren Bedeutungen kombinieren lässt. In der Phonographie dagegen besitzen die Zeichen lautliche Werte, die so in Sequenzen kombiniert werden können, dass sie (mehr oder weniger direkt) den Lauten gesprochener Wörter entsprechen. Phonographische Systeme kommen dabei in der Regel mit einem deutlich geringeren Zeichenvorrat aus.

²¹wozu Schriftzeichen sprichwörtlich „in Stein gemeißelt“ wurden

Weitergehende Erläuterungen zur Historie und feingranulareren Systematisierung von Schriftsystemen und Schriftzeichen als deren Basis können z. B. in [Haa01] nachgelesen werden. Dort findet sich auch ein für die Verarbeitung von Texten wichtiger Hinweis: „Schriftsysteme operieren nach eigenen Prinzipien, die in partieller, aber nicht vollständiger Wechselbeziehung zu sprachlichen Strukturen stehen.“

Dieser Überblick deutet bereits auf einige Herausforderungen für die diskrete Repräsentation von geschriebener Sprache hin. Sollen Schriftzeichen, die auf physischen Medien abgetragen sind, digital repräsentiert werden, so wird für sie die Zuweisung zu einem konkreten numerischen Code notwendig. Dieser muss allen Kommunikationspartnern bekannt sein, ebenso wie die genaue Form, Bedeutung und somit „Identität“ des dadurch repräsentierten Zeichens. Für die Codierung eines Schriftsystems ist daher die Auflösung von graphisch existierenden Mehrdeutigkeiten sowie eine Idealisierung von Symbolen als Zusammenfassung hinreichend ähnlicher stilistischer Varianten notwendig, um einen diskreten Satz von Zeichen festzulegen.

In der Anfangszeit der elektronischen Datenverarbeitung wurden Daten auf Lochkarten gespeichert, wie im Exkurs zu den Wurzeln des *Humanities Computing* bereits erwähnt wurde. Entsprechend des vorrangigen Einsatzgebiets im nordamerikanischen Raum haben sich zuerst Zeichensätze für das Lateinische Alphabet herausgebildet, wobei anfangs zunächst keine Unterscheidung in Groß- und Kleinschreibung vorgenommen wurde. Das Hinzufügen von weiteren Zeichen zum präferierten Zeichensatz ließ Formate mit sieben Bit langer Kodierung (und damit 128 unterscheidbaren Zuständen) entstehen, bis später eine Repräsentation mit acht Bit (also genau einem Byte) genutzt wurde. Die damit abgedeckten Zahlen waren über verschiedene Codetabellen dann konkreten Alphabeten zugeordnet. Die Angabe der Codetabelle war dabei nicht Bestandteil des Textinhalts und somit in den meisten Fällen nicht eindeutig aus dem Inhalt ersichtlich.

Erst zu Beginn der 1990er Jahre wurde mit Unicode ein System für das *Encoding* aller erdenklichen Alphabete und Schriftzeichen in einer gemeinsamen Codetabelle eingeführt. Dabei wurde ein universellerer Ansatz der Repräsentation von der Graphemen und Graphemgruppen, welche als Buchstaben verstanden werden, genutzt. Basis der Kodierung ist eine konzeptionelle Trennung in Glyphen (*glyphs*), Zeichen (*characters*) und abstrakte Zeichen (*abstract character*). Ein Überblick über die teils recht komplexe Kategorisierung und die damit verbundene Nomenklatur findet sich im „*Glossary of Unicode Terms*“²².

²²<http://unicode.org/glossary/>

Mit dem **Unicode Transformation Format (UTF)** wurde eine Familie von konkreten Binärkodierungen für die Codepunkte von Unicode-Zeichen eingeführt, die teils eine variable Bytelänge aufweisen und dennoch in Sequenz geschrieben und eindeutig ausgelesen werden können. Für die Speicherung einzelner Zeichen werden dabei bis zu 32 Bit verwendet. Übliche Varianten von Buchstaben, die meist durch das Hinzufügen sogenannter „Diakritischer Zeichen“ entstehen, können oft als Einzelzeichen, daneben aber auch stets als sequenzielle Kombination aus einer Basisglyphe und dem Zusatzzeichen kodiert werden. Für diese Fälle enthält Unicode vier Normalformen, wovon die wichtigsten die **Normalization Form Canonical Composition (NFC)** und die **Normalization Form Canonical Decomposition (NFD)** sind. Beide sind „kanonisch äquivalent“²³, stehen also trotz unterschiedlicher bitweiser Darstellung auf logischer Ebene für die selben Zeichen und sollten bei Zeichenkettenvergleichen (auf dieser Ebene) als identisch gewertet werden.

Darüber hinaus ergibt sich durch die binäre Speicherung und die diskrete Kodierung, dass Zeichenähnlichkeit sich nicht aus numerischer Ähnlichkeit der Repräsentation ableiten lässt. Auch in uneindeutigen Fällen muss jedoch eine der Kodierungsvarianten ausgewählt werden, weshalb z. B. die Suche nach Wörtern mit einem speziellen Zeichen in beliebiger Variante, die jeweils anderen nicht einschließt. In Abschnitt 3.6 auf Seite 91 wird die damit verbundene Problematik der Normalisierung noch ausführlicher dargelegt. Insbesondere zeitliche Aspekte und die damit verbundene Evolution von Schrift erzeugt uneindeutige Situationen bezüglich Form und Verwendung einzelner Zeichen. Dabei können in der Zeichentabelle immer nur klar abgrenzbare Verwendungskontexte einfangen werden. Die Entwicklung des römischen V zum gerundeten U, deren Doppelung zum W, dessen Transformation zum diakritischen hochgestellten Kleinbuchstaben ^w etc. sind in genau diesen festen Stufen abbildbar – nicht jedoch ihr Kontinuum von Zwischenschritten. Das Beispiel ist aus „Unicode Explained“ [Kor06] entlehnt, wobei die Problematik universeller ist und z. B. auch analog für Ligaturen und daraus entstehende Buchstaben, wie das ß oder veraltete Buchstabenvarianten, wie f (das lange „s“ im Deutschen) umfasst.

Vor der Standardisierung von Unicode wurden für die Eingabe und Speicherung von Texten in nicht-lateinischen Alphabeten zum Teil auch nicht-standardisierte Hilfskonstrukte geschaffen. Ein Beispiel ist die Verwendung von sogenanntem Betacode, einer systematischen Umdeutung lateinischer Zeichen zur „sekundären“ Kodierung antiker griechischer Texte: „Κάδμος“ wird darin z. B. als „*ka/dmos“ abgebildet.

²³siehe auch <http://unicode.org/reports/tr15/>

Während für diese Problematik mit Unicode eine umfassende und saubere Lösung geschaffen wurde (die jedoch eine komplette Neukodierung erfordert), enthält Unicode andererseits noch viele Relikte aus früheren Zeichentabellen, mit z. T. recht kurioser Bedeutung. An prominenter Codeposition Nummer Sieben findet sich z. B. ein Sonderzeichen namens „Bell“, das für eine akustische Meldung (Glockenschlag) anstatt eines lesbaren, gedruckten Zeichens steht – eine Praxis, die bis in die Zeit der Fernschreiber zurückreicht. Auch weitere funktionale Codepunkte regeln Ausgabemodalitäten, so etwa der Wagenrücklauf und der Zeilenumbruch, die (neben Leerzeichen) Funktionen für die Strukturierung von Text haben. Diese Strukturbildung als wichtiger Aspekt der Textrepräsentation soll im Folgenden noch eingehender untersucht werden.

2.2.3 Repräsentation der Struktur von Text

In diesem Buch werden Texte als Analyseeinheiten angesehen, welche aus diskret beschreibbaren Einzelteilen bestehen, die selbst wiederum als Analyseeinheiten dienen können. Mit dieser Sichtweise wird ein eher technisches Modell von Text entwickelt, um Methoden der Informatik auf Texte anwenden zu können. Natürlich umfasst der Textbegriff viele weitere Ebenen, welche als Lesarten (*Readings*) und tiefer greifende Interpretationen bei einer manuellen Analyse zum Ausdruck gebracht werden können. Für die Erfassung von Texten in digitalen Systemen bedarf es jedoch einfacher und präziser Beschreibungen, die dabei möglichst wenige Grundannahmen enthalten sollten.

Der Inhalt von Texten ergibt sich über die Abfolge von einzelnen Schriftzeichen und Leerstellen. Über diese können in vielen Sprachen Wörter als sinntragende Einheiten gebildet werden, welche ggf. noch von Interpunktion umgeben sind. Darüber können (je nach Verwendungsmuster der Schriftsprache) noch größere Sequenz-Abschnitte, wie Sätze, abgegrenzt und so direkt oder mittelbar aus der Einzelzeichenfolge abgeleitet werden.

Die Sprache selbst besitzt eine komplexe, teils durchaus mehrdeutige Struktur, die Produkt der kulturellen Entwicklung der Menschheit sowie der Historie einzelner Sprechergruppen ist und die seit der Antike Untersuchungsgegenstand der „Grammatik“ ist. Heute betreibt die Linguistik vielschichtige und nicht immer zueinander kompatible Forschungen zu den Strukturen von Sprache. Die Abbildung solcher Strukturen wird im Abschnitt [2.2.6 auf Seite 45](#) bei der Beschreibung von Annotationen vertieft. Hier soll zunächst nur die zusätzliche Strukturierung innerhalb von Texten (als linearisierte

Form von Sprache) betrachtet werden, welche sich als „beabsichtigte“ Strukturierung (vgl. „*Intentional Structure*“ in [GMPM13]) auffassen lässt. Diese tritt im Wesentlichen als Abschnittsbildung der oben beschriebenen Sequenzen hervor.

In Sequenzen existieren lokale Kontexte: Zeichen folgen auf andere Zeichen und Wörter befinden sich in der Nachbarschaft anderer Wörter. Sätze stehen im Kontext vorheriger (und nachfolgender) Sätze. Die Bildung größerer Einheiten (Zeichen, Wort, Satz, Absatz, Kapitel, etc.) hat jeweils den Zweck, die natürlichen lokalen Kontexte an geeigneter Stelle absichtlich zu unterbrechen.

Die Entscheidung über solche Trennungen inhaltlich schwächer zusammenhängender Teile ist auf den kleinteiligeren Ebenen stark sprachabhängig. Zum Beispiel ist die Trennung von Mehrwortbegriffen (*Multiword Units*) im Englischen gängige Praxis und im Deutschen eher die Ausnahme. Auch die Untergliederung von Wortfolgen in Sätze ist nicht beliebig möglich, sondern in großem Maße den Strukturen der Sprache geschuldet. Auf höherer Ebene können Abschnittsbildungen als Strukturierungsmittel jedoch im Wesentlichen frei verwendet werden. Konkret können inhaltliche Gründe oder stilistische Entscheidungen für die Untergliederung sprechen. Jedoch sind oft auch die Limitierungen und Eigenheiten des physischen (oder digitalen) Mediums ausschlaggebend für eine Untergliederung von Texten – das Platzangebot auf den Seiten eines Buches, das Format von Karteikarten oder Formularvordrucken, die symmetrisch zu beschriftenden Seiten des Sockels einer Statue, die Prägeflächen von Münzen, zeichenzahlbeschränkte Kurzmitteilungen und noch vieles mehr. Oft ist ein Zusammenspiel von „logischen“ als auch „physischen“ Aspekten für die Untergliederung von Text verantwortlich.²⁴

Unterschiedliche Textgattungen bringen zum Teil ihre eigenen Strukturmerkmale mit, wie z. B. Verse (ggf. mit Zäsuren) und Strophen. Größere Texte erfordern oft eine geschachtelte Untergliederung in thematische Abschnitte, wie Kapitel und Unterkapitel. Noch größere Textwerke sprengen die Limitationen von Trägermedien, so dass etwa mehrbändige Buchveröffentlichungen entstehen. In [RMD96] werden weitere Beispiele für solche Strukturelemente unter der Bezeichnung *text objects* gegeben.

Sequenztrennungen unterschiedlicher Stärke markieren die Grenzen unterschiedlicher Hierarchiestufen der Strukturelemente. Dadurch entstehen Gruppierungseffekte für die

²⁴Stede und Suriyawongkul beschreiben in [SS09] neben der *Logical Structure* noch die sogenannte *Content Structure*, durch welche der „Kommunikative Zweck“ des Textes transportiert wird. Diese Form der Textstruktur als Ergebnis inhaltlicher Analysen wird im Kontext dieses Buches eher als eine Form von Annotation angesehen und daher an dieser Stelle nicht detaillierter besprochen.

Zuordnung kleinerer zu größeren Einheiten und alle diese zusätzlichen Ebenen tragen ein Stück zur Auflösung der (z. T. mediengeschuldeten) Linearität des Textflusses bei. Das Verständnis von Text als Mischung von Inhalten und Strukturinformationen ist wichtig bei der diskreten Repräsentation in digitalen Systemen. Sie führt nicht zuletzt auch Einheiten ein, auf die in der externen Kommunikation über den Text verwiesen werden kann.

Unter Berücksichtigung dieser Vorüberlegungen ist klar, dass über die im letzten Abschnitt vorgestellte Zeichenrepräsentation hinausgehend umfangreiche Methoden zur maschinenlesbaren Repräsentation der Strukturen von Texten geschaffen werden müssen, um sie für wissenschaftliche Analysen adäquat abzuspeichern.

Im Jahr 1990 untersuchten DeRose und Kollegen die Frage „*What is Text really?*“ [DDMR90] und erstellten dabei das bis dahin fortschrittlichste Textrepräsentations-Modell, welches Text als „*Ordered Hierarchy of Content Objects (OHCO)*“ auffasst. Dabei nutzen sie die Modellierungsparadigmen der (sich damals noch in intensiver Entwicklung befindlichen) *Standard Generalized Markup Language (SGML)*. Die technische Umsetzung von Textstrukturierung über die Auszeichnung des rohen Fließtextes (das sogenannte *Markup*) mit einem hierarchischen Tag-System ist nach wie vor übliche Praxis. Das damit verbundene Verständnis von Text als schriftlichen Inhalten in Zeichenkettenform, welche eine singuläre Hierarchie und eine festgelegte lineare Reihenfolge besitzen, ist bis heute eine populäre Vorstellung. Daran hat auch die bereits zwei Jahre nach dem Verfassen des oben genannten Artikels in [RMD96] vorgenommene Neubewertung des gewählten Modellierungskonzepts nichts geändert. Diese war unter dem Eindruck aufkommender Zweifel, insbesondere bezüglich der mangelnden Eignung des OHCO-Modells für den Umgang mit überlappenden Hierarchien notwendig geworden. Neben dem Aufzeigen von Gegenbeispielen, bei denen unifizierte Hierarchien nicht als geeignete Struktur-Repräsentation gelten können, wird dort auch noch viel grundsätzlicher dagegen argumentiert, dass sich eine umfassende, mehrere „Perspektiven“ (und damit Hierarchien) umfassende Strukturierung überhaupt in geeigneter Weise in eine OHCO umformen lässt (*decomposition*).

Während dieser Umstand noch vor einiger Zeit als praktisch weniger relevant eingestuft wurde (z. B. in [Wit04] für Anwendungen der Linguistischen Informationsmodellierung²⁵), kann er heute angesichts des Wunsches, immer mehr Annotationsebenen und Strukturierungsvarianten gleichzeitig digital abzubilden, nicht mehr ignoriert werden. Bereits

²⁵ „In der Praxis ergeben sich durch diese Restriktion relativ selten Probleme, da verschiedene Strukturen häufig in einer Hierarchie repräsentiert werden können.“

2010 schrieb Schmidt in [Sch10] unmissverständlich: „*Overlap is a serious problem in the encoding of cultural heritage texts*“

Hieraus wird auch deutlich, dass keine zwei Perspektiven auf die Kodierung von Textstruktur einander gleichen. Soll der Pluralismus von Strukturierungsansätzen für konkrete Anwendungsfälle bei der Schaffung allgemeingültiger Textmodelle berücksichtigt werden, erfordert dies einen intensiven inhaltlichen Austausch in der Forschungscommunity. Diese Aufgabe wird seit den 1980er Jahren von der [Text Encoding Initiative \(TEI\)](#) übernommen. Das Gremium entwickelt das gleichnamige Auszeichnungsmodell für die Kodierung von Textquellen. In [Bur14] wird die [TEI](#) zurecht als „*one of the longest-lived and most influential projects in the field now known as the Digital Humanities*“ beschrieben.

Die aktuellen „[TEI Guidelines P5 2.0](#)“, s. [TEIP5], umfassen 1636 Seiten, auf denen die verschiedensten Überlegungen und Probleme dokumentiert sind, die im Kontext der Abbildung von textuellem Quellenmaterial für den Einsatz in den e-Humanities relevant sind. Für eine Vielzahl an Editions-Methoden und -Szenarien wird ein Schema für die Erzeugung von [TEI](#)-Dokumenten in [XML](#), angegeben²⁶. Es wird also gewissermaßen ein Vokabular und die Grammatik vorgegeben, mit denen sich um maschinenlesbares Markup ergänzte Texte erfassen und verarbeiten lassen.

Angesichts des großen Umfangs des [TEI](#)-Standards mit unzähligen nicht standardisierten Kombinationsvarianten von Markup für die Modellierung von Sonderfällen wurden zahlreiche, „handlichere“ und strengere Unterformate erstellt. Das vom Deutschen Textarchiv konzipierte Basisformat²⁷, das vom Institut für deutsche Sprache entwickelte und mit [TEI](#)-Entwicklungen synchronisierte Textmodell²⁸, aber auch das von der [TEI](#) selbst entwickelte [TEI-Simple](#)²⁹ versuchen, eine Ausgangsbasis für einheitliche [TEI](#)-konforme Dokumente zu schaffen.

Als ganz ähnlich zu [TEI](#) ist die im [CTS](#) vertretene Auffassung der Struktur von Text als Hierarchie zitierbarer Einheiten anzusehen, die im Wesentlichen auch der [OHCO](#) folgt. Für kanonischen Text ist diese Sichtweise unbedenklich, da sie genau die Wissenschaftskultur widerspiegelt, in der eine Primärhierarchie herausgegriffen und als Zugangsform festgelegt wird. Diese kann auch als Grundlage für Editionen verwendet werden und gewährleistet die Möglichkeit zur standardisierten Referenzierung von Einheiten.

²⁶Frühere [TEI](#)-Versionen nutzen noch die [SGML](#).

²⁷<http://www.deutschestextarchiv.de/doku/basisformat>

²⁸<http://www1.ids-mannheim.de/kl/projekte/korpora/textmodell.html>

²⁹<http://github.com/TEIC/TEI-Simple>

Für andere, allgemeinere Anwendungsfälle häuft sich aber – wie bereits erwähnt – spätestens ab der Jahrtausendwende die Kritik an der Textrepräsentation als OHCO. Grundsätzlicher Konsens ist, dass die digitale Textrepräsentation das Vorhandensein eines Modells voraussetzt, wie z. B. Buzzetti in [Buz02] schreibt:

In and of itself, every representation and, consequently, every form of text representation entails the implicit or explicit assumption of a model, at least if we accept the postulate that the "map is not the territory".

Zur Definition eines adäquaten Modells für Text wird eine geeignete Abstraktionsstufe benötigt, welches die Defizite der OHCO-Sichtweise ausgleicht. Dafür wurde allerdings bis heute keine allgemeingültige Lösung gefunden.

In [Bra05] wird die Ergänzung von hierarchischen XML-basierten Textrepräsentationen um nicht-hierarchische Elemente vorgeschlagen, wobei die damit eingeführten Verknüpfungen und Querverweise stets als Modellierungskonstrukte „zweiter Klasse“ angesehen werden müssen, die über Standardwerkzeuge im XML-Umfeld nur unbefriedigend genutzt werden können. Daher gibt die gewählte Primärhierarchie immer noch die dominierende Modellierungs- und Analyse-Sichtweise vor. Auch die in Abschnitt 2.2.6 auf Seite 43 näher vorgestellten Methoden zur Annotation von Texten können grundsätzlich für Strukturauszeichnungen verwendet werden – in der Praxis ist dies jedoch ebenfalls mit Problemen und fehlender Werkzeugunterstützung verbunden, so dass diese Repräsentationsmöglichkeit selten genutzt wird.

Auch andere Formen der Verknüpfung von Strukturelementen wurden untersucht. HyTime z. B. arbeitet mit Hyperreferenzen und verfügt laut Selbstbeschreibung über „very effective constructs to support the basic hierarchical component structure of most documents“ [DD94]. Allerdings handelt es sich auch hierbei um ein frühes, SGML-basiertes Format ohne nennenswerte Werkzeugunterstützung. Generell soll zur Fokussierung des in diesem Buch beschriebenen Textmodells eine Abgrenzung gegenüber Hypertexten stattfinden. Hypertextualität – in schwacher Form wie bei Webseiten genutzt, bis hin zu ausgeklügelten Konzepten, wie beim Project Xanadu³⁰ – ist typischerweise (noch) kein breiter Forschungsgegenstand der e-Humanities. Zudem wurden moderne Formen von Hypertext beispielsweise in [Meh09] bereits mit graphbasierten Modellen beschrieben.

³⁰<http://www.xanadu.net/> – ein seit 1960 laufendes Projekt mit der Vision von Dokumenten, die Teile anderer Dokumente „virtuell“ enthalten können, inklusive einer Verbreitungs-, Lizenzierungs- und Abrechnungsinfrastruktur, vgl. [Nel93]

Die digitale Text- und Strukturrepräsentation dient, wie am Beispiel der [TEI](#) gezeigt, als explizites konzeptionelles Modell, aber auch als Speicherformat. In den weiteren Ausführungen werden solche Formen der technischen Textrepräsentation ausgeklammert, die kein allgemeineres Modell von Text beinhalten, etwa Strukturen wie *Text-Buffer* von Editoren zum effizienten interaktiven Editieren und Transformieren von Zeichenketten. Dagegen besitzen die ebenfalls eher technischen Ansätze in der automatischen Textanalyse, wie beschrieben, sehr große Relevanz für die beschriebenen Arbeiten. Die dort üblichen Repräsentationsformen werden daher gesondert im folgenden Abschnitt vorgestellt.

2.2.4 Textrepräsentation im Text Mining

Die digitale Textrepräsentation kann vielfältig geschehen und wird meist aufgabenorientiert festgelegt. In den informatiknahen Bereichen der e-Humanities steht die maschinelle Auswertung der Texte im Vordergrund: Das *Text Mining* stellt Werkzeuge für die Extraktion relevanter sachlicher und inhaltlicher Zusammenhänge aus digital vorliegenden Texten bereit, vgl. [[HGW08](#)]. Bei Texten handelt es sich (aus dieser Sichtweise betrachtet) im Grunde um unstrukturierte Daten, auf die statistische und regelbasierte Verfahren angewendet werden. Es handelt sich allerdings um eine spezielle Form des *Data Mining*, bei welchem die Besonderheiten natürlicher Sprache sowie heterogener und unstrukturierter Quellen berücksichtigt werden müssen. Es bewegt sich damit im Forschungsbereich des [Natural Language Processing \(NLP\)](#), also der automatischen Sprachverarbeitung. Computerlinguistische Forschung wird dabei nicht primär betrieben, wobei einzelne Ergebnisse oder digitale Werkzeuge aus diesem Forschungsbereich bei entsprechender Eignung durchaus in Text-Mining-Verfahren einfließen können.

Im Text Mining werden die in persistierter Form vorliegenden Daten üblicherweise transformiert und in einen anderen Verarbeitungs-Zustand überführt, bevor die eigentlichen Verfahren und Analysen zur Anwendung kommen. Übliche Schritte sind die Unterteilung und Diskretisierung der Zeichensequenzen in untersuchungswürdige Einheiten. Oft wird für diese aus Effizienzgründen eine (meist fortlaufende) numerische *Feature-ID* vergeben und damit von der buchstabengetreuen Repräsentationsform abstrahiert. Über die zu untersuchenden Analyseeinheiten werden dann häufig Matrizen gebildet, die das Vorkommen innerhalb größerer Einheiten ausdrücken. Dabei kann es sich um Strukturelemente mit logischer Entsprechung (wie im vorigen Abschnitt beschrieben) handeln,

aber auch um rein künstlich segmentierte Einheiten, wie „Buchstaben- n -gramme“³¹, die statistisch erfasst werden sollen.

Als theoretische Basis und methodischer Ausgangspunkt für die Analyse der Bedeutung von Wörtern (bzw. Wortbestandteilen) über Methoden der Statistik wird meist die Vorstellung von einer „Distributionellen Semantik“ genannt. Diese geht auf Überlegungen von Ferdinand de Saussure aus dem frühen 20. Jahrhundert zurück, die dem sogenannten Strukturalismus – einer Strömung in der Linguistik – zugerechnet werden.³² Die Bedeutung von Wörtern hängt demnach wesentlich von den Kontexten ab, in denen es verwendet wird – und somit auch von den Bedeutungen (und Kontexten) der gemeinsam mit ihnen auftretenden Wörtern.

Zur Bedeutungsanalyse werden daher oft kookkurrenzbasierte Verfahren herangezogen, die gemeinsames Auftreten von Wörtern in irgend geeigneter Form quantifizieren (und dabei meist eine Art statistischer Signifikanz erfordern). Auch Verfahren, die Wörter nach ihren Kontexten gruppieren, leiten sich direkt aus diesen Überlegungen ab. Durch diskrete Untersuchungseinheit der Wörter kann – wenn man diese als unabhängige Dimensionen betrachtet – ein Vektorraum aufgespannt werden. Ein detaillierter und sehr informativer Überblick über die so formulierbaren *Vector-Space-Modelle* wird in [TP10] gegeben³³.

Wird die Vektorrepräsentation von Wörtern für alle Kontexte gesondert vorgenommen, ergibt sich eine Matrix, deren einzelne Werte das Vorkommen des jeweiligen Wortes im jeweiligen Kontext anzeigen. Automatische Analysen der Wortbedeutung versuchen oft, eine kondensierte Kontextrepräsentation zu finden, aus der sich die Originalmatrix in möglichst ähnlicher Form rekonstruieren lässt, wobei verschiedene Formen der Dimensionsreduktion zum Einsatz kommen. Generell werden im Text Mining Verfahren aus verschiedensten Bereichen angewendet, wie der linearen Algebra, der statistischen Approximation, den künstlichen neuronalen Netzen oder der Optimierungsprobleme. Aktuell gilt einiges Forschungsinteresse den Arbeiten zu sogenannten *Word Embeddings*. Deren Entwicklung kann in [Sah15] detailliert nachgelesen werden. In diesem Artikel werden insbesondere auch die konzeptionellen Beiträge von einigen Wegbereitern des Text Mining, wie Zellig Harris, John Rupert Firth und Ludwig Wittgenstein, angesprochen.

³¹ n -gramme sind Teilsequenzen der Länge n – Buchstaben- n -gramme sind also alle Teilzeichenketten, die aus n nacheinander in der Originalzeichenkette auftretenden Buchstaben gebildet sind.

³²Diese eine Einschätzung wird mittlerweile z. T. auch hinterfragt, wie etwa in [Jäg07].

³³Wobei die dort geäußerte alleinige Zuschreibung der grundlegenden Idee zu Gerard Salton von anderen Autoren bezweifelt wird, vgl. [Dub04].

Diese hier umfassend nachzuzeichnen, würde den Rahmen des Buches jedoch sprengen.

Beim Erfassen von Kontexten wird häufig der sogenannte *Bag-of-Words*-Ansatz gewählt, der innerhalb der Kontexte reine Frequenzen (also Anzahlen) betrachtet und dabei die Reihenfolge von Wörtern ignoriert. Um die syntaktische (speziell nach de Saussure „syntagmatische“) Struktur der Texte dennoch berücksichtigen zu können, geschieht neben der Betrachtung von Wörtern teils auch eine Bildung von Wort-n-Grammen. Auch eigene Repräsentationsformen für gleichzeitige Abbildung von Wortbedeutung und Reihenfolgeinformationen ist möglich, wie z. B. in [JM07] durch ein „zusammengesetztes holographisches Lexikon“.

Inhaltsbestimmung, Ähnlichkeitsbestimmung und Gruppierung von Dokumenten sind eng miteinander verwandt. Die oben angesprochene Dimensionsreduktion des Auftretens von Wörtern in Kontexten kann auch genutzt werden, um die Kontexte zu vergleichen, wobei Dokumente (wenn sie in einer homogenen Kollektion vorliegen) natürlich geeignete Kontexte für die Analyse darstellen. Entsprechend haben sich mit dem *Latent Semantic Indexing* (bzw. *Latent Semantic Analysis*) [DDL⁺90], dessen probabilistischer Variante [Hof99] und den auf bayesscher Statistik beruhenden Topic Models – allen voran der „*Latent Dirichlet Allocation*“ [BNJ03] – Verfahren entwickelt, die die Zusammensetzung von Dokumenten aus (wenigen) latenten „Bedeutungsklustern“ von Wörtern postulieren und diese aus Daten errechnen. Für Aufgaben des *Information Retrieval*, also des Auffindens relevanter Dokumente angesichts eines Informationsbedürfnisses liefert diese kondensierte Repräsentationsform eine geeignete zusätzliche Entscheidungshilfe. Die klassischen Formen der Textrepräsentation im Information Retrieval sowie Formen der Indizierung von Dokumenten in Kollektionen werden noch eingehender in Abschnitt 3.7.1 auf Seite 95 beschrieben. Eine Abgrenzung zu erweiterten „Dokument“-Begriffen wird im folgenden Abschnitt vorgenommen.

2.2.5 Dokumentrepräsentation

Als Dokumente werden üblicherweise Einheiten textueller Inhalte bezeichnet, die oft in weitestgehend statischer Form vorgehalten werden. Klassische Dokumente sind auf physischen Trägern festgehaltene Texte, die dort in visueller Form mitsamt ihrer Struktur ablesbar sind, wozu das Verständnis des gewählten Layouts (z. B. zweispaltige Artikel mit eingerückten Zitaten) erforderlich ist. Digitale Dokumente sind Sammlungen von Einzelinformationen, aus denen sich bei Bedarf ein (elektronisch angezeigtes oder gedrucktes)

Dokument ableiten lässt. Welche Prozesse an dieser Ableitung beteiligt sind und wie explizit das Dokument Informationen über die spätere visuelle Dokumentenerscheinung enthält, ist dabei sehr stark von Technologie und dem jeweiligen Einzelfall abhängig.

Bei E-Books liegt der Fokus neben dem reinen Text zusätzlich auch auf der Repräsentation der Textstruktur und der dokumentbegleitenden Metadaten. Die konkrete Anzeige richtet sich jedoch in der Hauptsache nach den Einstellungen und (physischen) Eigenschaften des E-Book-Readers, wie präferierter Schriftart, Displaygröße oder der Bildschirmausrichtung bei tragbaren Endgeräten. In dieser Hinsicht sind E-Books vergleichbar mit Dokumenten, die in der im [World Wide Web \(WWW\)](#) verwendeten [Hypertext Markup Language \(HTML\)](#) verfasst sind. Diese unterstützt Querverweise (*Hyperlinks*) zwischen [WWW](#)-Ressourcen, aber auch zwischen Dokumentteilen. Darüber hinaus erlaubt sie die Einbindung von Bildern, Audio- und Videodateien über die [Uniform Ressource Locator \(URL\)](#) genannte Webadresse einer solchen Mediendatei. In [HTML](#) wird das Aussehen im Wesentlichen über sogenannte [Cascading Style Sheets \(CSS\)](#) definiert, in denen die Layoutinformationen deklarativ auf Grundlage der Dokumentstruktur notiert sind.

[HTML](#) ist ein ehemals [SGML](#)-basiertes Markupformat, das zwischenzeitlich in [XML](#) neuformuliert wurde, bevor es nun wieder losgelöst davon weiterentwickelt wird. Im Umfeld von [XML](#) existieren mit der [XML Stylesheet Language \(XSL\)](#) weitere Werkzeuge zur Umformung von strukturierten inhaltstragenden Dokumenten in alternative (ggf. um Layoutinformationen ergänzte) Repräsentationsformen. Die Erstellung anzeigbarer Dokumente aus [XML](#)-basierten Daten- oder Textsammlungen geschieht durch Transformation (gegebenenfalls auch erst auf Anfrage – „on-the-fly“) über die [XSL Transformation \(XSLT\)](#). Mit den [XSL Formatting Objects \(XSL-FO\)](#) steht ein Formatierungen tragendes Dokumentenformat für diesen Prozess zur Verfügung. Auch gängige Büroanwendungen verwenden zum Teil [XML](#)-basierte Formate, in denen Inhalt, Struktur, Layout und Metadaten in auf den Funktionsumfang und die Bedürfnisse des jeweiligen Programms abgestimmter Form abgebildet werden. Beim Prozess der [Optical Character Recognition \(OCR\)](#), bei dem aus Bilddateien digitalisierter Druckerzeugnisse maschinenlesbare Texte generiert werden, wird ebenfalls eine um Aspekte des Dokumentlayouts erweiterte Repräsentationsform benötigt. In der Praxis wird dafür z. B. das [HTML](#)-basierte Format „hOCR“³⁴ verwendet.

Abseits der auch menschenlesbaren Formate hat sich aus dem Bereich der technischen Buchherstellung (der sogenannten Druckendstufe) heraus das [Portable Document Format \(PDF\)](#) als ein effizientes Binärformat etabliert. Dieses wird seit Längerem auch für die

³⁴<http://kba.github.io/hocr-spec/1.2/>

elektronische Betrachtung und den digitalen Austausch von Dokumenten eingesetzt. Wegen weithin fehlender Möglichkeiten, Änderungen an den Dokumenten vorzunehmen, wird ihnen gemeinhin ein gewisser „Urkundencharakter“ zugesprochen, was z. B. ihren Einsatz für bestimmte Zwecke der Verwaltung befördert hat.

Wie gezeigt, existieren zahlreiche Formate und Konzepte zur anzeigenahen Repräsentation von Dokumenten. Das Erfassen von Informationen über Layout, physische Beschaffenheit und Aussehen von Dokumenten ist jedoch nicht Gegenstand dieses Buches, da das Recherchesystem auf eine semantische Repräsentation der Texte ausgerichtet sein soll. Zwar sind, etwa in [Aud08], bereits Ansätze beschrieben worden, die maschinenlesbare Repräsentationsformen und Analysemodi insbesondere für komplexe Dokumente entwickeln³⁵ – noch sind diese jedoch nicht Gegenstand breiter Forschung. Zudem bringen sie einen stark erhöhten Komplexitätsgrad für die Auswertung mit sich, der hier zunächst vermieden werden soll.

2.2.6 Repräsentation von Metadaten und Annotationen

Die bereits in Abschnitt 2.1.3 auf Seite 25 vorgestellte inhaltliche, strukturelle und fachspezifische Vielfalt der zu erfassenden Metadaten führt dazu, dass sich für diesen Aspekt der Datenverarbeitung viele unterschiedliche Lösungsansätze entwickelt haben. Bevor einige davon genauer vorgestellt werden, soll zunächst noch einmal kurz erörtert werden, worum es sich bei Metadaten überhaupt handelt. Die informelle Beschreibung als „Daten über Daten“ greift in vielen Anwendungsszenarien deutlich zu kurz. Ob z. B. eine Autorenangabe ein Metadatum ist, hängt von der konkret eingenommenen Perspektive ab. Für einen Datenbestand, in dem Zeitungstexte verwaltet werden, ist sie eine (in vielen Fällen verzichtbare) Zusatzinformation. In bibliographischen Datensätzen ist der Buchautor dagegen Teil der verwalteten Kerninformation: „Daten über Bücher“ – nicht „über Daten“. Der Teilbegriff der Meta-„Daten“ ist zudem etwas unglücklich gewählt, da Daten meist eine sehr rohe und wenig interpretierbare Einheit darstellen, aus der ohne Kenntnis ihrer genauen Struktur keine Information abgeleitet werden kann. Im Rahmen dieser Arbeit soll im Hinblick auf diese beiden Aspekte allerdings keine allzu strenge Abgrenzung erfolgen. Alle „zusätzlichen“ Informationen in einer Wissensbasis, deren schematische Bedeutung erklärbar ist³⁶, werden hier als Metadaten bezeichnet.

³⁵wie etwa digitalisierte Handschriften, Notizen, Skizzen, Logbücher, Collagen usw.

³⁶Das heißt, es soll bekannt sein, welchen Aspekt ein Metadatum beschreibt. Die genaue Bedeutung des eingetragenen Wertes muss dabei nicht näher spezifiziert (oder bekannt) sein.

Im Kontext dieses Buches handelt es sich bei Metadaten meist um allgemeine Zusatzinformationen zu einem Text oder zu größeren Teilen davon. Damit besitzen sie eine gewisse Nähe zu Markup, worüber wieder eine enge Verknüpfung mit der Strukturrepräsentation entsteht. Beispielsweise ist die Metainformation „Das Buch ist in fünf Kapitel gegliedert.“ vollständig aus der Struktur ableitbar.

Textbezogene Metadaten sind das wichtigste Zugangsinstrument zu digitalisierten Textkollektionen. In [Nun09] wird in diesem Zusammenhang massive Kritik an der oberflächlichen Digitalisierungsweise im Google-Books-Projekt³⁷ geübt, welches große Defizite hinsichtlich der Qualität (insbesondere Vollständigkeit und Korrektheit) der erfassten Metadaten aufweist. Solche ungenügend kuratierten Zugriffsmechanismen behindern die wissenschaftliche Arbeit mit Korpora enorm.

Metadaten sollten im Idealfall maschinenlesbare Inhalte besitzen und eine angemessene digitale Repräsentation der jeweiligen Zusatzinformationen darstellen. Für viele Querschnittsaspekte, wie Ort und Zeit sowie Angaben zu deren jeweiligen Granularitäten und der Unsicherheit seitens des Erfassers konnte jedoch noch kein allgemeingültiger Repräsentationsstandard erstellt werden. Auch für fachbezogene Aussagen fällt es oft schwer, einheitliche Standards für Metadaten zu etablieren. Um eine Einigung hinsichtlich der Bedeutung konkreter Metadatensätze zu erreichen, bietet es sich an, sogenannte Ontologien zu modellieren, eine Form semantischer Netze, auf die später noch genauer eingegangen wird. Die Modellierung kann dabei ausgehend von grundsätzlichen Überlegungen zur konzeptuellen Einteilung der Anwendungsdomäne (*top down*) oder ausgehend von der Gruppierung einzelner Phänomene und Datensätze (*bottom up*) erfolgen. Für die Etablierung standardisierter Ontologien ist in der Regel eine inhaltliche und organisatorische Zusammenarbeit in größeren Gremien und Konsortien notwendig.

Im Bibliothekssektor haben sich Metadatenmodelle wie das [MACHINE-Readable Cataloging \(MARC\)](#) und das [Metadata Object Description Schema \(MODS\)](#) herausgebildet. Viele Institutionen haben dabei eigene Erweiterungen an diesen Modellen vorgenommen, stimmen jedoch im Kernbereich mit dem Einheitsmodell überein. Allgemeine Referenzmodelle, wie das bereits angesprochene [CIDOC-CRM](#), ermöglichen eine konzeptuelle Zusammenführung von Modellen unterschiedlicher, aber verwandter Ressorts, etwa von Archiven und Museen oder Bibliotheken und digitalen Textrepositorien. Für die dadurch ermöglichte Nutzung von Infrastrukturen stellt sich die Frage, ob aus allen Teilmodellen ein einziges zentralisiertes Modell erzeugt werden soll, oder die Infrastruktur eine logische

³⁷<http://books.google.de/>

Verknüpfung dezentralisierter Metadatenrepositorien vornimmt. In CLARIN kommt mit der [Component Metadata Infrastructure \(CMDI\)](#) eine Technologie für letzteres Szenario zum Einsatz, die nachnutzbare Schema-Komponenten für die Definition eigener Metadatenformate bereitstellt. Auf diese Weise gelingt es in der föderierten Umgebung gleiche Metadatenteile zusammenzuführen und unterschiedliche Teile als solche zu isolieren und für spezifische Abfragen dennoch nutzbar vorzuhalten.

Annotationen sind eine spezielle Form von Metadaten, die für kleine Ausschnitte der Daten, wie Textstellen, Bildregionen, Einzeltonfolgen usw. notiert werden. Auch sie sind eng mit dem Begriff des Markup verbunden. Es zeigt sich in der Praxis jedoch häufig eine leicht unterschiedliche Ausrichtung: Annotationen sagen meist etwas über einen (externen) digital abzubildenden Text aus, während mittels Markup üblicherweise eher deklariert wird, wie ein bestimmter Textteil (durch eine Maschine) zu interpretieren ist. Renear kritisiert in [Ren01] im Hinblick auf die TEI grundsätzlich die Unterscheidung von Markup in die Kategorien *descriptive* und *procedural*. Schmidt weist in [Sch14] auf das „TEI-Paradox“ hin, indem er aufzeigt, dass, obwohl die XML-basierte Technologie auf vollständige Interoperabilität ausgelegt ist, ein einzelnes TEI-kodiertes Dokument auf der Ebene der Bedeutung von Tags nicht interoperabel ist.

Auch durch solche Erkenntnisse hat sich die Praxis des Notieren von Annotationen direkt im Dokument (*inline*) schrittweise zur Nutzung von losgelösten Annotationen (*stand-off*) gewandelt, vgl. z. B. [BW09]. Stand-Off-Annotation können, wie bereits angesprochen, auch zur Auszeichnung von Struktur verwendet werden. Für alle Anwendungsgebiete benötigen sie eine eindeutige Positionierungsmöglichkeit in den Dokumenten, um Beginn und Ende einer Annotation korrekt numerisch angeben zu können. Für den Positionierungs-Offset werden genaue Angaben zur Zählweise benötigt – graphemweise, codepunktweise, bytewise, wortweise oder über per Markup definierte Anker.

Annotationen können durch automatische Verfahren vorgenommen werden oder durch manuelle Bearbeiter ergänzt werden. Während im Bereich der e-Humanities durch große Korpora oft sehr umfangreiche Annotationsaufwände anfallen, die nur selten mit automatischen Hilfsmitteln in ausreichender Qualität bewältigt werden können, ist in diesem Bereich für die manuelle Annotation nicht immer tiefes Fachwissen nötig. Dies eröffnet Möglichkeiten für die Beteiligung von fachfremden Freiwilligen, in Form von *Citizen-Science*-Projekten und in *Crowdsourcing*-Umgebungen, wobei ein mehrfaches Annotieren der gleichen Stellen durch verschiedene Bearbeiter zur Vermeidung sporadischer Fehler angeraten ist.

Nachdem die Komplexität der Text- und Korpusrepräsentation nun auf verschiedenen Ebenen vorgestellt wurde – angefangen bei der Codierung einzelner Zeichen bis hin zur Repräsentation von heterogenen Sammlungen annotierter Dokumente – soll in den nächsten Abschnitten untersucht werden, welche Technologien für Speicherung und Abfrage solcher Daten zur Verfügung stehen. Ausgehend von der Transition der traditionell relationalen Datenbanklandschaft zu einem Ökosystem aus einer Vielzahl neuer Ansätze und Systeme sollen insbesondere die Paradigmen zum Umgang mit hochverknüpften Datensammlungen herausgegriffen werden, denn: Eine flexible Textrepräsentation benötigt flexible Formen der Verknüpfung von Daten.

2.3 Graphdatenbanken

2.3.1 NoSQL-Datenbanken

Um die aktuellen Entwicklungen im Datenbanksektor besser einordnen zu können, bedarf es zunächst eines kurzen Rückblicks auf die Historie dieser Technologie. Einen lesenswerten, weil sehr breiten (und entsprechend wenig tiefgreifenden) Abriss über relevante Entwicklungen gibt z. B. [LC13]. An dieser Stelle sollen nur kurz Schlaglichter auf einzelne relevante Entwicklungsschritte geworfen werden:

Während der Bedarf an Speicherung und Prozessierung großer Mengen von Daten bis in die Anfangszeit der Rechnersysteme zurückreicht – die Geschichte des *Index Thomisticus* ist dafür nur ein Beispiel von vielen – so waren die damaligen Rechnersysteme nicht mit unserer heutigen Vorstellung von Datenbanksystemen vergleichbar. Eingabe und Ausgabe erfolgten rein sequenziell, ein Zugriff auf einzelne *Records* war zwar bei Karteikarten theoretisch noch möglich, jedoch bei Magnetbändern nicht effizient (und damit ökonomisch vertretbar) zu bewerkstelligen.

Erst mit dem Aufkommen von wahlfreiem Zugriff auf gespeicherte Dateien nach der Erfindung von Festplatten wurden Aufgaben der individuellen Datenspeicherung und Abfrage relevant. Als Analogie für die neuen Zugriffsmechanismen wurde (auch aus Gründen der Kontinuität) hauptsächlich die Karteikarte und die Tabelle gewählt. Formelle mathematische Beschreibungen wurden in Form der Relationalen Algebra eingeführt, während sich in der Praxis **Create, Read, Update, Delete (CRUD)** als die vier Grundoperationen für persistente Daten etablierten.

Mit wachsendem Verständnis der mathematischen Implikationen und mit steigender Zahl an Anwendungsszenarien wurde der Fokus dann auf die Formulierung und Prüfung von Konsistenzbedingungen für die Datenbestände gelegt. Zur Vermeidung von Redundanzen – durch die nicht wenige Inkonsistenzen hervorgerufen werden können – wurde der Fokus auf die Forschung an sogenannten Normalformen für die Datenmodellierung gelegt. Zu Beginn der 1970er Jahre wurde die bis heute relevante, auf früheren Arbeiten aufbauende (und teils zu vorhergehenden Definitionen äquivalente) Boyce-Codd-Normalform³⁸ vorgestellt, vgl. [Cod74]. Um die Konsistenz der Daten auch in Mehrbenutzerszenarien mit konkurrierenden Zugriffsmustern zu gewährleisten, wurden Transaktionskonzepte und grundsätzliche Systemanforderungen, wie [Atomicity, Consistency, Isolation, Durability \(ACID\)](#) definiert, vgl. [HR83].

Das relationale Datenbankmodell war zu diesem Zeitpunkt bereits das am häufigsten verwendete und es wurden viele Werkzeuge und Vorgehensmodelle für den Umgang damit geschaffen, etwa das [Entity-Relationship Model \(ERM\)](#) zur abstrakten (graphischen) Formulierung von Daten- und Domänenmodellen, vgl. [Che76]. Mit der Standardisierung der [Structured Query Language \(SQL\)](#) als herstellerübergreifende Abfragesprache wurden die relationalen Datenbanken der Quasi-Standard für Datenhaltung. Da parallel zu diesen Entwicklungen in großer Zahl Arbeitsplätze mit Bürorechnern ausgestattet wurden und immer mehr Planungs-, Verwaltungs- und Kommunikationsaufgaben in digitalen Systemen erledigt wurden, konnten sich diese Datenbanksysteme als Speicher-Backend für eine Vielzahl von Fachanwendungen durchsetzen, wodurch das Marktvolumen für Datenbanktechnologien noch einmal enorm anstieg.

Seitdem wurden kommerzielle Weiterentwicklungen hauptsächlich innerhalb des dadurch abgesteckten Technologierahmens umgesetzt und es konnten viele (auch aus Sicht der Informatik wertvolle) Optimierungen umgesetzt werden – allerdings waren seitdem wenige „die Grundfesten erschütternde“ Neuerungen zu verzeichnen. Im Geschäftsbetrieb haben sich im Wesentlichen zwei Anwendungsfälle herauskristallisiert, die auch in technologischer Hinsicht zu unterschiedlichen Produkten geführt haben. Datenhaltung und Abfrage im Umfeld von Geschäftsanwendungen verfolgen entweder die Strategie des [On-line Transaction Processing \(OLTP\)](#) oder die des [On-line Analytical Processing \(OLAP\)](#), s. [HW05]. [OLTP](#) setzt den Fokus auf interaktive Abfragen, während [OLAP](#) die Möglichkeit für aufwändige Analysen auf dem gesamten Datenbestand in einer verzögerten Ausführung bietet.

³⁸nach Raymond F. Boyce und Edgar F. Codd

OLAP-Systeme, die auch als *Data Warehouses* bezeichnet werden, gewinnen zunehmend an Bedeutung für die Forschung, wenn Auswertungen im Bereich von Big Data notwendig sind. Bei sehr großen Datenmengen ist eine Verteilung auf mehrere physische Rechnersysteme notwendig, wobei es (aus der Theorie ableitbare) grundsätzliche Einschränkungen im Bezug auf die Leistungsfähigkeit des Gesamtsystems im verteilten Szenario gibt. Das als **Consistency, Availability, Partition tolerance (CAP)** bekannte Theorem besagt, dass die Konsistenz der verteilten Daten (C), die Verfügbarkeit und Geschwindigkeit des Systems (A) sowie dessen Toleranz gegenüber Ausfällen und Störungen (P) nicht gleichzeitig optimal sein können, sondern maximal zwei davon.

Diese Ausgangssituation einer omnipräsenten, weitestgehend statischen Datenbanktechnologie, die vor erhebliche Skalierungsprobleme gestellt war, bildete den Nährboden für unabhängige Neuentwicklungen. Hauptsächlich im Rahmen von *Open-Source*-Projekten wurde an Alternativen zu relationalen Datenbanken gearbeitet, bis eine Vielzahl solcher Ansätze schließlich in einer Initiative namens **Not only SQL (NoSQL)** zusammengefasst wurde. Ausdrücklich wenden sich die dort gebündelten Projekte nicht gegen relationale Datenbanken an sich, sondern gegen die Monokultur, die von diesen begründet wurde und durch welche Auswahlmöglichkeiten beschränkt wurden. Nicht jeder Anwendungsfall benötigt **ACID**-Eigenschaften, nicht jeder Anwendungsfall benötigt alle **CRUD**-Operationen und die Abwägungen, die hinsichtlich der **CAP**-Eigenschaften getroffen werden müssen, sollten auch vom Anwendungsfall abhängig gemacht werden.

NoSQL-Systeme umfassen unter anderem:

- *Key-Value Stores*, die für numerische Schlüssel oder Schlüssel aus Zeichenketten meist primitive Werttypen speichern und oft Abfragen über Wertebereiche oder Teilzeichenketten der Schlüssel zulassen,
- *Wide Column Stores*, die für die Kombination aus Zeilenschlüssel und Spaltenschlüssel (für potentiell sehr viele Zeilen und Spalten) Werte hinterlegen können
- Dokumentdatenbanken, in denen geschachtelte Schlüssel-Wert-Container als „Dokumente“ gespeichert werden, z. B. in der **JavaScript Object Notation (JSON)**,
- Graphdatenbanken (und Hypergraphdatenbanken), in denen zwei (oder mehr) Einträge, welche Schlüssel-Wert-Paare enthalten, direkt miteinander in Beziehung gesetzt werden können, ohne dass diese Assoziation auf der Ebene von Klassen oder Tabellen definiert werden muss und

- *Tuple Stores* für die Speicherung prädikatbasierter Ausdrücke (sogenannte „Tripel“ aus Subjekt, Prädikat und Objekt). Soll über diese Aussagen wiederum etwas ausgesagt werden, müssen sie in referenzierbarer Form als Quadrupel gespeichert werden. Um diese zu referenzieren, werden Quintupel verwendet, ...

Darüber hinaus werden teils auch spezialisierte Systeme – im textorientierten Bereich beispielsweise [XML](#)-Datenbanken³⁹ und Volltext-Suchmaschinen⁴⁰ – die über entsprechende Speicherfunktionen verfügen, zu den [NoSQL](#)-Systemen gerechnet.

Die meisten dieser Systeme zeichnen sich durch weniger rigide Anforderungen an die Definition eines Datenbankschemas aus, als es bei relationalen Systemen üblich ist – oder bieten überhaupt keine Möglichkeit zur Schemadefinition. Grundsätzlich gilt, dass je komplexer die abbildbaren Strukturen sind, umso weniger Daten für effiziente (komplexe) Abfragen im Echtzeitzugriff vorgehalten werden können. Daneben kann als Merkmal vieler Systeme eine Unterstützung für *Sharding*-Mechanismen und den Betrieb auf Rechnerclustern festgestellt werden. In diesem Zusammenhang werden teils eigene Lösungen, zum anderen aber nicht selten auch Komponenten aus dem Big-Data-Ökosystem um Apache Hadoop⁴¹ genutzt, welches als eine der ersten Ausführungsumgebungen für MapReduce [[DG04](#)] frei zur Verfügung steht.

Die dabei oft angewendete Form der Systemskalierung unter Aufweichung oder gänzlicher Aufgabe der strengen [ACID](#)-Bedingungen wird unter der Bezeichnung [Basically Available, Soft state, Eventually consistent \(BASE\)](#) in [[Pri08](#)] wie folgt eingeführt:

BASE is diametrically opposed to ACID. Where ACID is pessimistic and forces consistency at the end of every operation, BASE is optimistic and accepts that the database consistency will be in a state of flux. Although this sounds impossible to cope with, in reality it is quite manageable and leads to levels of scalability that cannot be obtained with ACID

In der Anwendung wird allgemein eine „elastische“ Skalierung bevorzugt. Die Elastizität bezieht sich auf die Fähigkeit zur Skalierung des Systems von einem physischen Einzelrechner zu großen (in wenigen Minuten bedarfsgerecht anmietbaren) Rechnerclustern, im laufenden Betrieb und ohne dass dafür initiale Vorkehrungen nötig sind.

³⁹z. B. eXistdb (<http://www.exist-db.org/>) und BaseX (<http://basex.org/>)

⁴⁰z. B. Elasticsearch (<http://www.elastic.co/de/products/elasticsearch>) und Apache Solr (<http://lucene.apache.org/solr/>)

⁴¹<http://hadoop.apache.org/>

Die **NoSQL**-Initiative hat viele technologische Neuerungen und Rekonzeptualisierungen im Datenbankbereich mit sich gebracht, oft jedoch zum Preis fehlender Anschlussfähigkeit und Kompatibilität zu bestehenden Fachanwendungen. Allerdings kann sich mittlerweile auch in der Welt der **NoSQL**-Ansätze für Datenmodellierung die Abfrage über **SQL** wieder positionieren. **SQL** kann über eine intermediäre Zugriffsschicht für verschiedene Datenspeicher genutzt werden, wie sie etwa in Apache Drill⁴² umgesetzt ist, s. [HN13]. Beim kommerziellen Datenbanksystem NuoDB⁴³ wird eine skalierbare Systemarchitektur mit **SQL**-Funktionalität z. B. als „NewSQL“ beworben.

Der im Zentrum dieser Arbeit stehende Begriff des Graphdatenbanksystems kann eine Vielzahl verschiedener Nuancen dieser Technologie bezeichnen. Der Praxis aus [RWE13] folgend,⁴⁴ werden hier damit vorrangig „Property-Graph-Datenbanken“ bezeichnet, deren Datenmodell später noch ausführlicher besprochen wird. Zunächst sollen die Besonderheiten von Graphen – jenen komplexen Strukturen, die sich in solchen Systemen abbilden lassen – in den folgenden Abschnitten vorgestellt werden.

2.3.2 Netzwerke, Graphen und ihre Anwendungsgebiete

Die Beschreibung und Analyse von Netzwerken lässt sich keiner einzelnen Disziplin zuordnen. Zum einen sind diese Aktivitäten oft Teil interdisziplinärer Forschung, zum anderen bilden sie aus methodischer Sicht einen eigenständigen Querschnittsaspekt der Forschung insgesamt. Eine mit vielen Beispielen und historischen Meilensteinen unterfütterte Einführung in Netzwerke und Netzwerkforschung sowie deren Bedeutung für unser tägliches Leben gibt Barabási in [Bar02].

Erste Aufmerksamkeit erhielt die Netzwerkforschung durch sozialwissenschaftlich geprägte Studien. Mitte der 1960er Jahre berichtete Milgram in [Mil67] von seinen Experimenten zur (oft erstaunlich kurzen) Pfadlänge zwischen Personen über (indirekte) Bekanntschaften. Diese Sichtweise hatte eine große Ausstrahlung in die breite Öffentlichkeit und es entwickelte sich daraus später die Vorstellung von *six degrees of separation*, über welche (angeblich) alle Teile der Weltbevölkerung miteinander verbunden seien.

Wenig später wurden ebenfalls aus soziologischer Motivation heraus wichtige Beiträge

⁴²<http://drill.apache.org/>

⁴³<http://www.nuodb.com/>

⁴⁴wohl wissend, dass deren Autoren großes wirtschaftliches Interesse an der Förderung genau dieses Graphdatenbank-Typs haben

im Bereich kleiner Netzwerke und direkter Interaktionen zwischen Personen geleistet. Prominentestes Beispiel ist die von Zachary durchgeführte Untersuchung der persönlichen Relationen zwischen den Mitgliedern eines Karateclubs von 1970 bis 1972, s. [Zac77]. Die sozialen Interaktionen und persönlichen Spannungen wurden festgehalten und es konnte exemplarisch nachgewiesen werden, dass Sie zum Zeitpunkt der Abspaltung eines neuen Karateclubs einen entscheidenden Einfluss auf den Verbleib oder die Abwanderung einzelner Mitglieder hatten. Daraus entwickelten sich nach und nach fortgeschrittenere Methoden der Netzwerkanalyse, die gemeinsam mit anderen Modellen in der Soziologie bis heute Anwendung finden.

In die selbe Zeit fallen auch allgemeinere Untersuchungen zu topologischen Aspekten großer Netzwerke, die sich bis in die heutige Forschung fortsetzen. Speziell zum Verhältnis von Strukturen auf der Mikroebene zu Effekten auf der Makroebene wurden z. B. in [Gra73] Überlegungen zu „*Weak Ties*“ veröffentlicht. Diese sind eine Form topologisch induzierter und grundsätzlich eher kontraintuitiver Effekte, die besagen, dass der größte Einfluss auf wesentliche Elemente der Netzwerkstruktur nicht von stark miteinander verknüpften Regionen, sondern von einzelnen, potentiell fragilen, Querverknüpfungen ausgeht.

Neben der Soziologie sind auch Physik, Biologie, Informatik, Logistik, Energiewirtschaft und Telekommunikation wichtige Impulsgeber für die Entwicklung der Netzwerkforschung zu nennen. Noch vor zehn Jahren war zu bemängeln, dass verschiedene Forschungs- und Anwendungsgebiete der Netzwerkanalyse sich untereinander nur mangelhaft abstimmen, von den Resultaten der anderen Bereiche kaum profitieren und folglich auch eine stark heterogene Forschungslandschaft bilden⁴⁵, vgl. [Jac06]. In seitdem neu entstandenen Feldern, wie etwa den „Computational Social Sciences“, vgl. [LPA⁺09], wird versucht, die bestehenden Methoden anderer Disziplinen (so weit dies sinnvoll umzusetzen ist) in den Forschungsprozess zu übernehmen.

Es lassen sich viele Arten von Netzwerken unterscheiden. Neben Untergliederungen nach Anwendungsgebiet kann eine Klassifizierung auch über die Art der abgebildeten Aussagegegenstände erfolgen: So kann es materielle, immaterielle oder hypothetische Objekte oder Akteure enthalten und die Verbindungen zwischen Ihnen können als Ähnlichkeit, Nähe, Affinität, Erreichbarkeit oder als konkrete Wege für den Fluss von Gütern, Werten oder Information angesehen werden. Eine allumfassende Einteilung mit genauer Dokumentation für die damit verbundenen Implikationen hinsichtlich einer Auswertung ist schwierig und liegt bislang nicht vor.

⁴⁵ „[...] they are still largely distinct in their methods, interests, and goals.“

Während die Bezeichnung „Netzwerk“ oft einen stärkeren Anwendungsbezug ausdrückt, hat sich für eine eher abstrakte, technische und auf die reine Struktur ausgerichtete Sicht auf Netzwerke die Bezeichnung „Graph“ etabliert. Graphen und Netzwerke sind Modelle verschiedener Abstraktionsstufen, die eine systematische Sicht auf Zusammenhänge in einer Anwendungsdomäne bieten können. Die Überführung von theoretischen Überlegungen zu dieser Domäne in passende Modelle und einzelner Beobachtungen in diskrete Datensätze innerhalb dieses Modells wird auch als *Graph Induction* bezeichnet. Auf die Graphinduktion in konkreten Anwendungsszenarien wird später in diesem Buch noch häufiger eingegangen.

Graphstrukturen finden sich nicht zuletzt auch häufig in der Informatik, etwa in Form von Listen, Bäumen und vielen komplexeren Datenstrukturen. Letztlich lassen sich auch die Gesamtheit der Objektinstanzen und -referenzen in der objektorientierten Programmierung als Graph ansehen, was insbesondere für die automatische Speicherbereinigung (*Garbage Collection*) von großer Relevanz ist. Ein unverzichtbares Werkzeug zur formellen Beschreibung solcher abstrakter Strukturen wird im folgenden Abschnitt vorgestellt.

2.3.3 Formalisierung und graphentheoretische Zugänge

Die Graphentheorie ist eine im 18. Jahrhundert wurzelnde Forschungsrichtung, die im Laufe des 19. Jahrhunderts systematisch mit anderen Teilen der Mathematik verknüpft wurde. Die ersten Kernprobleme und frühen Entwicklungen als Disziplin sind z. B. in [BLW79] ausführlich dargelegt. Die Graphentheorie wurde erst spät, im Jahr 1936 mit [Kön36] – der ersten diesbezüglichen Veröffentlichung in Buchform – als eigenständiges Wissensgebiet etabliert. Seitdem wurden innerhalb der Mathematik, aber auch durch die Informatik, viele neue Erkenntnisse zu Graphen gewonnen. Gleichzeitig half die Beschreibung als Graphen-Problem auch beim Finden neuer Lösungsansätze für verschiedene externe Fragestellungen. Nicht selten fand dabei ein methodischer Übertrag in andere Gebiete statt – so etwa in die natur- und sozialwissenschaftliche Forschung, wie bereits im letzten Abschnitt beschrieben.

Im Kontext von Graphdatenbanken ist die reine Graphentheorie allerdings nur ungenügend als Formalismus geeignet. Der Autor schließt sich der in [RN11] geäußerten Kritik an einer beinahe „reflexhaften“ Angabe eines übersimplifizierten Graphen-Formalismus am Anfang vieler diesbezüglicher Veröffentlichungen an. Wie Rodriguez und Neubauer weiterhin feststellen, ist in praktischen Anwendungsszenarien (die vielleicht im Gegen-

satz zur Mathematik in der Informatik häufiger anzutreffen sind) die Realisierung einer graphförmigen Struktur und die Interaktion mit ihr oft wichtiger, als die formelle Analyse ihrer graphentheoretischen Eigenschaften.

Problematisch ist weiterhin, dass die schon angesprochene Universalität des Graphen-Begriffs dazu beigetragen hat, dass eine Vielzahl intuitiv entstandener Definitionen parallel existieren, die nicht immer kompatibel zueinander sind. Eine Einführung in die heute gebräuchlichen Formalisierungen für Graphen geben z. B. [Die16] und [GS12]. An dieser Stelle sollen die Kernbegriffe zunächst kurz informell vorgestellt werden:

Bei der Beschreibung von Graphen existieren zwei zentrale Konzepte: Ein **Knoten**⁴⁶ kann Objekte der realen Welt bzw. menschlichen Vorstellung repräsentieren oder einfach als anonymes und bedeutungsloses Element einer Menge angesehen werden. Eine **Kante**⁴⁷ verbindet zwei Knoten miteinander.

Der Graph ergibt sich im einfachsten Fall aus einer Menge von Knoten und einer Menge von Kanten. Die Graphentheorie nutzt zur Definition dieser Grundelemente und ihrer Eigenschaften Konstrukte der Mengentheorie – üblicherweise jedoch, ohne auf eine bestimmte Axiomatisierung näher einzugehen. In Anbetracht der in [Lei14] vorgebrachten Kritik⁴⁸ kann dies durchaus problematisch sein: Generell beantwortet die Graphentheorie nicht die Frage, welche (mathematischen) Konstrukte die Knoten eines Graphen eigentlich konkret darstellen. Für Graphdatenbanken, bei denen direkt in Knoten Informationen persistent gespeichert werden sollen, ist diese Unterspezifikation besonders ungünstig.

Für die sehr einfache Definition von Graphen existieren zahlreiche Erweiterungen: Gerichtete Graphen, in denen bei Kanten die Reihenfolge der verbundenen Knoten beachtet wird; gewichtete Graphen, deren Kanten um Zahlwerte ergänzt sind; Hypergraphen, deren Kanten mehr als zwei Knoten verbinden; Multigraphen, die mehrere Kanten zwischen den selben Knoten erlauben, und noch vieles mehr. Allerdings verändern all diese Variationen auch die mathematischen Konstrukte in der Graphen-Definition. Die Aussage „Ein gerichteter Graph ist ein Graph.“ ist daher strenggenommen (je nach konkreter Definition) entweder nicht formell untersuchbar oder gar falsch! Darüber hinaus existieren verschiedene (nicht in allen Belangen äquivalente) Möglichkeiten der konkreten

⁴⁶auch: „Ecke“, englisch: *node* oder *vertex*

⁴⁷auch: „Bogen“, englisch: *edge*

⁴⁸Kurz zusammengefasst: Beim „traditionellen“ Axiomensystem für Mengen, „Zermelo-Fraenkel with Choice“, sind alle Elemente von Mengen selbst Mengen. Da die Identität von Mengen sich über ihre Elemente ergibt, können „Primitive Elementtypen“, wie Zahlen, prinzipiell nicht einfach in diesem System verwendet werden.

Ausgestaltung der Erweiterung, etwa um für gerichtete Graphen die Richtung einer Kante anzugeben: Bei Diestel wird ein ungerichteter Graph um zwei Abbildungsfunktionen (von Kanten auf Knoten) ergänzt, die jeweils Start- und Zielknoten bestimmen. Andere Veröffentlichungen und Einführungswerke definieren bei gerichteten Graphen die Kantenmenge als Menge von Mengen der Mächtigkeit 2 zu Mengen von Paaren um (so z. B. auch Griffin), oder beschreiben die Änderungen überhaupt nicht in formaler Form. Ähnlich heterogen werden in der Literatur auch die anderen angesprochenen Erweiterungen gehandhabt.

Neben der Graphentheorie existieren weitere Zweige der Mathematik bzw. der theoretischen Informatik, die sich mit Graphen beschäftigen, etwa die (allgemeine sowie endliche) Modelltheorie und die Kategorientheorie [BW95]. Diese Betrachtungsweisen erlauben grundsätzlich eine Verallgemeinerung auf andere Arten von Strukturen, wodurch sich neue Anwendungsmöglichkeiten ergeben. Es wurde z. B. in [SK12] bereits gezeigt, dass die Kategorientheorie auch zur Wissensrepräsentation geeignet ist. Daneben existieren weitere mathematische Zugänge, wie die spektrale Graphentheorie, die tatsächlich nützliche Aussagen für die Abfrage sehr großer graphförmiger Daten in Datenbanken liefern kann, wie z. B. in [ZCYL08] festgestellt wurde. Dennoch stellen auch diese Formalisierungen im Umfeld von Graphdatenbanken keine geeigneten Zugänge zur mathematischen Beschreibung der Gesamtsysteme dar.

Aus der klassischen Sichtweise der Informatik ergeben sich zum Umgang mit Graphen jedoch vor allem Fragestellungen der digitalen Repräsentationsform. Dabei werden im Bereich der Algorithmen und Datenstrukturen einführend vorrangig Adjazenzmatrizen, Adjazenzlisten und Inzidenzmatrizen sowie Knoten- und Kantenlisten betrachtet. Diese anschaulichen und gut verstandenen Repräsentations- und Speicherformen haben allerdings so gut wie keine praktische Relevanz im Datenbankumfeld. Spinrad gibt in [Spi03] einen Überblick über „*Efficient Graph Representations*“. Doch letztlich sind auch solche Betrachtungsweisen beschränkt, da sie sich jeweils fast ausschließlich nur auf einfache strukturelle Aspekte der zu speichernden Elemente konzentrieren, während Datenmodellierung, -speicherung und -abfrage sich nicht unmittelbar und für die anwendungsnahe Forschung gewinnbringend in den klassischen Formalismen der Graphentheorie abbilden lässt. Im Umfeld von Graphdatenbanken lassen sich jedoch andere, passendere formelle Beschreibungen finden, wie im nächsten Abschnitt noch erläutert wird.

2.3.4 Property-Graph-Datenbanken

Graphdatenbanken sind keine Erfindung der letzten Jahre. Allerdings sind sie erst durch die Dynamik der [NoSQL](#) Bewegung wieder in den Fokus von Forschung und Industrie geraten. Warum ihre spezialisierte Herangehensweise an die Modellierung von Daten und deren Relationen nicht früher breitere Anwendung gefunden hat, lässt sich nur schwer abschließend ergründen. In [[Bue12](#)] etwa heißt es dazu:

The research of graph databases was popular in the early 1990s with database models like LDM, GOOD, O2, and GraphDB. However, this interest died off with the resurgence of XML and the Internet. Not until recently have graph databases again become a topic of interest. This re-emergence is due in part to the large amounts of graph data introduced by the Web.

Diese Beschreibung wirkt in sich recht widersprüchlich: Das [WWW](#) hat gerade durch seine emergente Verlinkungsstruktur schnell große (netzwerkförmige) Datenmengen produziert und sich nicht zuletzt durch dieses „verbindende“ Merkmal als prominentester Service im Internet etabliert. Um die Strukturen des [WWW](#) detailliert zu erforschen, wären Graphdatenbank-Systeme bereits in den 1990er Jahren hilfreich gewesen.

Ein möglicher Aspekt, der zur zwischenzeitlichen Abwendung von der Forschung an Graphdatenbanken geführt hat, könnte im großen Entwicklungsvorsprung relationaler Systeme (bei relativ ähnlichem Funktionsumfang) begründet liegen. Beim unvoreingenommenen Vergleich des relationalen Modells mit Graphenmodellen lassen sich im Grunde nur wenige fundamentale Unterschiede ausmachen:

Zum einen betrifft dies die Schematisierung der Daten: Einträge sind nicht mehr in Relationen zusammengefasst, über die die ihnen zuweisbaren Eigenschaften definiert werden, ähnlich wie über Klassen in der objektorientierten Programmierung. Stattdessen kann für jeden einzelnen Eintrag eine beliebige Menge an Eigenschaftsschlüsseln verwendet werden. Zum anderen betrifft es die Art der Indizierung von Werten. Neben globalen (in relationalen Systemen tabellenzentrischen) Indizes stehen zusätzlich also knotenzentrische (*vertex centric*) Indizes zur Verfügung, wie in [[RN11](#)] detaillierter vorgestellt wird. Verknüpfung wird so zum „Modellierungskonstrukt erster Klasse“, über das sich effizient und ohne die Notwendigkeit von *Join*-Operationen zwischen Datensätzen navigieren lässt. Jedoch ist laut [[SFSK15](#)] auch eine effiziente Emulation einer Graphdatenbank über das relationale Modell möglich.

Letztlich können die Unterschiede auf technischer Ebene (anders als aus konzeptioneller Sicht) demnach nicht wirklich fundamental sein. Auch etablierte Modellierungswerkzeuge, wie das [ERM](#) sind mit nur geringen Modifikationen für die Nutzung mit Graphdatenbanken geeignet. Datenmodelle zum Speichern von vernetzten Daten kommen in verschiedenen Ausprägungsformen zum Einsatz. In [\[Bue12\]](#) und [\[Ang12\]](#) werden die Modellierungsansätze und -konstrukte gängiger Graphdatenbank-Systeme beschrieben und verglichen. Im Wesentlichen bilden diese Auflistungen aus dem Jahr 2012 auch die noch heute genutzten Modell-Varianten ab.

Property Graphs stellen dabei eine recht kleinteilige und minimalistische Sicht auf Einträge und Zusammenhänge zwischen ihnen dar. Sie haben sich zu einem populären Modell entwickelt, für das auch kommerzielle Software entwickelt wird. In ihrer jetzigen Form sind sie als eine Art „Industrie-Standard“ anzusehen, wobei mit der von der Apache Foundation verwalteten Programmbibliothek TinkerPop⁴⁹ eine quasi-normative Instanz existiert. Die dort enthaltene Schnittstellenbeschreibung „Blueprints“ definiert eine gemeinsame [Application Programming Interface \(API\)](#) für viele Graphdatenbanken unterschiedlicher Hersteller.

In [\[JV13\]](#) wird das Property-Graph-Datenmodell als *directed, edge-labeled, attributed, multi-graph* beschrieben⁵⁰. In Multi-Graphen können die selben zwei Knoten über mehrere verschiedene Kanten verbunden sein. Die Kanten in Property-Graphen sind stets gerichtet und besitzen einen Kantentyp (*Edge Label*). Property-Graphen sind zudem „attributiert“, Knoten und Kanten besitzen Eigenschaften (*Properties*⁵¹), die in Form von Schlüssel-Wert-Paaren hinterlegt werden. Die Schlüssel von Eigenschaften sind (wie auch die Kantenlabels) Teil des Schemas der Graphdatenbank. Das Datenbankschema schränkt nicht ein, welchen Knoten oder Kanten welche Eigenschaften zugewiesen werden dürfen, oder welche Kantenlabels in welchen Kontexten verwendet werden dürfen. Diese Restriktionen sind durch die Anwendungslogik vorzunehmen und gegebenenfalls bei der Abfrage zu berücksichtigen.

Im Folgenden wird ein Vorschlag für einen geeigneten Formalismus zur Abbildung der wesentlichen Merkmale von Property-Graphen entwickelt, der sich auf die Herange-

⁴⁹<http://tinkerpop.apache.org/>

⁵⁰wobei der dort dafür angegebene Formalismus (offenbar aus Gründen der Übersichtlichkeit) weder Kantenlabels noch Attribut-Wert-Paare incl. deren Zuweisungsfunktionen zu Kanten bzw. Knoten und Kanten berücksichtigt

⁵¹nicht zu verwechseln mit *graph properties*, also den mathematisch bestimmbaren Eigenschaften von Graphen im graphentheoretischen Sinne, etwa „enthält ein Dreieck (K^3) als Subgraph“, vgl. [\[Die16\]](#)

hensweise und Symbolbezeichnungen in [RN11] stützt, in welchem jedoch Schema und Instanzdaten getrennt betrachtet werden:

Ein Property-Graph-Schema Γ ergibt sich aus einer Menge Σ von Kantenlabeln und einer Menge R von Property-Schlüsseln:

$$\Gamma = (\Sigma, R)$$

Ein Property-Graph, der diesem Schema folgt, ergibt sich aus einer Menge V von Knoten, einer Menge E von Kanten, einer Menge P von Property-Feldern, einer Menge S von Property-Werten, einer Kanten-Label-Funktion λ und einer Property-Wertzuweisungsfunktion μ :

$$G_\Gamma = (V, E, P, S, \lambda, \mu)$$

Die Kanten sind gerichtet:

$$E \subseteq (V \times V)$$

Die Property-Felder stehen für Property-Schlüssel, die in einzelnen Elementen (Knoten und Kanten) des Graphen vorkommen:

$$P \subseteq (V \cup E) \times R$$

Jeder Kante wird genau ein Kantenlabel als „Kantentyp“ zugewiesen:

$$\lambda : E \rightarrow \Sigma$$

Jedem Property-Feld wird genau ein Property-Wert zugewiesen⁵².

$$\mu : P \rightarrow S$$

In der Praxis wird die Anwendbarkeit von Property-Graphen wesentlich von der Indizierung der Elemente bestimmt. Bei dieser wird meist eine Unterscheidung in Knoten- und Kantenproperties getroffen. Im Schema würde das eine Unterteilung in R_V und R_E erfordern, die entsprechende Änderungen in der Definition des Graphen nach sich zöge, indem dort in μ_V und μ_E unterschieden werden müsste, welche wiederum mit entsprechend definierten $P_V = V \times R_V$ und $P_E = E \times R_E$ zu versehen wären. Bei der Indizierung

⁵²In der aktuellen Version 3 von Apache TinkerPop können grundsätzlich für die selben Schlüssel zu den selben Elementen multiple Werte zugewiesen werden, was in dieser Arbeit jedoch nicht genutzt und noch nicht von allen Systemen unterstützt wird. Deshalb soll dieser Umstand zunächst auch keinen Eingang in diesen Formalismus finden.

erfolgt zudem auf Schemaebene eine Zuweisung von Datentypen oder Wertebereichen zu Property-Schlüsseln, die zusätzlich eine Segmentierung von S nach diesem Kriterium erfordern würden.

Da in diesem Buch der Formalismus aus den im letzten Abschnitt erläuterten Gründen nicht mehr aufgegriffen wird, wird an dieser Stelle auf eine allzu ausführliche formelle Beschreibung verzichtet. Ebenso wird darauf verzichtet, den Formalismus von Rodriguez und Neubauer für die Graphentraversierung (als Operationen auf Multigraphen) aufzugreifen.

Bevor in Abschnitt 2.3.6 auf Seite 62 noch eine detaillierte Übersicht über Abfragemöglichkeiten und -sprachen für Property-Graphen gegeben wird, sollen zunächst ergänzende Bemerkungen zu einem Gebiet erfolgen, das eng mit den bisher besprochenen Themen der Markupsprachen, Ontologien, Netzwerke, Textkollektionen und Datenbanken verwandt ist.

2.3.5 Semantic-Web-Technologien

Angeichts der erstaunlich schnellen und weitgreifenden Popularisierung des [WWW](#) hin zum meistgenutzten Service im Internet, war es kurz nach der Jahrtausendwende an der Zeit, neue Visionen für eine alltagstaugliche, global vernetzte Technologie zu schaffen. Berners-Lee, Hendler und Lassila stellten im Jahr 2001 im Artikel [\[BLHL01\]](#) das „*Semantic Web*“ die nächste „Evolutionsstufe“ des Webs vor, das bis dahin nur universelle (und somit unspezifische) Verknüpfungen zwischen genauso universellen (und unspezifischen) Informationsressourcen ermöglichte. Mit der expliziten Modellierung von Bedeutungen einzelner Aussagen sollte es damit möglich werden, eine logische Verbindung verteilt vorliegender Aussagen (auch) durch Maschinen vornehmen zu lassen.

Als Grundlage für die maschinenlesbare Formulierung von Bedeutung wurden Tripel aus Subjekt, Prädikat und Objekt gebildet. Ersteres steht für eine Ressource, also einen Aussagegegenstand, dessen Identität durch einen *Identifier* gekennzeichnet werden kann. Prädikate bilden semantische Ankerpunkte für Eigenschaften, welche das genaue Verhältnis von Subjekt und Objekt spezifizieren. Das Objekt kann ein einfacher Wert eines bestimmten Datentyps sein (eine Ganzzahl, ein Datum oder eine Zeichenkette), kann aber auch (über deren Identifier) auf eine andere Ressource verweisen. Auf diese Weise kann aus mehreren Aussagen ein Graph erstellt werden. (Entsprechend sind auch die [NoSQL](#) -

Technologien der Graphdatenbanken und Triple Stores im Grunde recht verwandt.)

Zur formalisierten Ressourcenbeschreibung wird im Semantic Web das [Resource Description Framework \(RDF\)](#) genutzt. In diesem werden Zeichenketten als Identifier verwendet, die den Anforderungen an den Aufbau eines [Uniform Resource Identifier \(URI\)](#) genügen. Die nutzbaren Prädikate bilden dabei das sogenannte „Vokabular“. Das Schema, das Aussagen über die grundsätzliche Validität (nicht den Wahrheitsgehalt) möglicher Aussagen angesichts eines Vokabulars tätigt, wird als [RDF Schema \(RDFS\)](#) abgebildet⁵³. Hierbei zeigt sich ein starker Zusammenhang zu Ontologien (im informatischen Sinne). In [[HFBPL09](#)] werden diese so eingeführt:

An ontology consists of statements that define concepts, relationships, and constraints. It is analogous to a database schema or an object-oriented class diagram. The ontology forms an information domain model.

Es wird dort weiter darauf verwiesen, dass solche Ontologien für viele Anwendungsgebiete bereits existieren und diese auf einfachem Wege (unverändert) nachgenutzt oder angepasst werden können. Im Kontext des Semantic Web wird für die Beschreibung von Ontologien die [OWL Web Ontology Language \(OWL\)](#) verwendet. Als Ontologiesprache besitzt sie eine höhere Komplexität und höhere Ausdrucksstärke als [RDFS](#). Wegen der starken Bindung des Begriffs an diese Technologie wird im Folgenden die Bezeichnung Ontologie eher sparsam verwendet und stattdessen öfter von Daten- und Domänenmodellen gesprochen.

Das Semantic Web bringt eine Sammlung von Basisbedeutungen für Klassen von Aussagegegenständen und für grundlegende Beziehungen mit sich. Über diese ist bereits eine sogenannte Inferenz möglich. Das Inferieren von Aussagen bedeutet: „[...] *given some stated information, we can determine other, related information that we can also consider as if it had been stated.*“ [[AH08](#)] So sind z. B. Aussagen über Superklassen auch für die Subklassen gültig, ohne dass dies explizit notiert werden muss.

Darauf aufbauend kann ein *Reasoning* erfolgen, also eine logische Folgerung der Gültigkeit von Ausdrücken, welche mittels [OWL](#)-Konstrukten der Beschreibungslogik oder in einer Regelsprache notiert sind. Da das Semantic Web eine verteilte Wissensumgebung darstellt, werden beim Umgang mit Daten nach diesem Paradigma eine Reihe impliziter Annahmen getroffen. Am prominentesten ist die sogenannte *Open World Assumption* im

⁵³[RDFS](#) nutzt dabei [RDF](#)-Modellierungskonstrukte, das Schema ist also Teil der Daten

Bezug auf im Datenbestand nicht enthaltene Aussagen: In Datenbanken gelten nicht auffindbare Einträge als inexistent, so dass damit verknüpfte, die Existenz des Eintrags voraussetzende Aussagen folglich als falsch interpretiert werden. Für eine Behandlung des Semantic-Web-Datenbestandes mit Konstrukten der elementaren Logik wäre ein solches Verhalten nicht zielführend, da aus einer falschen Prämisse bekanntermaßen jede beliebige Aussage gefolgert werden kann. Die *Open World Assumption* geht also davon aus, dass die Aussage existieren könnte – und nur momentan nicht im Datenbestand zur Verfügung steht. Weitere Details zu in den Modellen gemachten Annahmen sowie konkrete Anwendungsbeispiele für [RDF](#)-Vokabulare und in [OWL](#) modellierte Ontologien können zum Beispiel in [\[HFBPL09\]](#) nachgelesen werden.

Aufbauend auf der Logik-Fähigkeit der Technologien sieht das Semantic Web auch die Beweisbarkeit der Korrektheit von Domänenmodellen (*Proof*) sowie Schichten für das Herstellen von Vertrauen (*Trust*) in Datenprovider, ebenso wie Querschnittsaspekte, wie Verschlüsselung vor. In diesen Bereichen des sogenannten *Semantic Web Stack* sind bislang jedoch keine breit adaptierten Entwicklungen vorgenommen worden. Im Bereich maschinenlesbarer Semantik wurde bereits vor Aufkommen der Semantic-Web-Technologie an Inferenzmechanismen gearbeitet, beispielsweise durch die Verknüpfung von [SGML](#) und [XML](#) mit Prolog, vgl. [\[SMHR00\]](#). Jedoch sind auch diese Ansätze nicht weiter verfolgt worden.

Das Semantic Web definiert sich nicht nur durch seine Technologien, sondern in erster Linie auch über die Wissensbasen und verteilten Ressourcen, die damit bislang realisiert wurden. Als wichtigster Ankerpunkt für die Semantik von Begriffen und Entitäten aus dem Bereich des enzyklopädischen Weltwissens kann das Projekt DBpedia⁵⁴ angesehen werden, welches mittels eines hauptsächlich automatischen Transformationsprozesses strukturierte Informationen aus der Wikipedia in [RDF](#) umwandelt, s. [\[BLK⁺09\]](#), bzw. ausführlicher [\[LJ⁺15\]](#).

Die Daten von DBpedia wurden auf dem Portal Freebase⁵⁵ mit weiteren Datenquellen verknüpft und nach dessen Übernahme durch Google in deren Technologie zur semantischen Suchunterstützung namens „*Knowledge Graph*“⁵⁶ überführt. Mittlerweile sind die (teils auch in Freebase durch Nutzereingaben angereicherten) Daten Teil des WikiData-Projekts⁵⁷. Freebase nutzte „graphd“ als Datenbank, eine Eigenentwicklung, die in

⁵⁴<http://dbpedia.org/>

⁵⁵<http://freebase.com/>

⁵⁶<http://www.google.com/intl/bn/insidesearch/features/search/knowledge.html>

⁵⁷<http://www.wikidata.org>

[MDGM10] näher beschrieben ist. Es handelt sich dabei um ein Speichersystem für Tupel unterschiedlicher Länge ohne physische Löschoption (append-only).

Diese Projekte sind Beispiele für den Trend zur leichtgewichtigen Nutzung von Semantic-Web-Technologien, ohne komplexe Ontologie- und Logikmodelle dafür zu definieren. Bei einer solchen Herangehensweise wird von Linked Data, bzw. bei offenen, frei verfügbaren Datenquellen von [Linked Open Data \(LOD\)](#) gesprochen. Über die gemeinsame Nutzung von Identifiern können verteilte Wissensspeicher ohne direkte Kommunikationsaufwände „aufeinander referenzieren“. Es formiert sich bei entsprechend sorgfältiger Auswahl von Identifiern ein Netz von auf Instanzebene verknüpften Datenquellen, die im Idealfall viele gemeinsame und etablierte Vokabulare verwenden.⁵⁸

Während im Semantic Web der Fokus auf semantisch ausgezeichneten strukturierten Informationen liegt, existieren Arbeiten, die eine enge Verzahnung mit unstrukturierten Informationsquellen anstreben, s. etwa [Lad13] zum Umgang mit hybriden Datenbeständen aus textuellen Quellen und strukturierten Informationen mittels [RDF](#)-Technologien. Das Semantic Web besitzt dabei eine große Nähe zu Markupsprachen. So existieren Ansätze zur Nutzung der Graphstruktur von [RDF](#) für eine verbesserte Repräsentation von markupbasierten Texten, z. B. mit „[Extremely Annotational RDF Markup \(EARMARK\)](#)“, vgl. [DIPV11], wo eine enge Anlehnung an Formate für *Wordprocessing*-Software stattfindet. Andererseits kann [XML](#) z. B. auch als Serialisierungsform für [RDF](#) und [OWL](#) verwendet werden und viele der im Semantic Web verwendeten Techniken entstammen direkt oder indirekt dem Umfeld von [XSL](#) und [HTML](#). Dabei ist die Nutzung von Markupsprachen nicht für alle Anwendungsfälle unumstritten, wie z. B. [SET09] anmerkt:

One of the major criticisms of semantic web formats like RDF/XML is that they are too complicated and too much of a hassle for a designer or webmaster to bother implementing.

Als Alternative für Web-Publisher wird dort weiter beschrieben, wie sich [RDF](#)-Prädikate in Form sogenannter Mikroformate oder unter Nutzung von [HTML](#)-Attributen in Webseiten ausdrücken lassen. Auch neue Strömungen innerhalb der Linked-Data-Community sagen sich langsam von [XML](#) und textuellem Markup als Austauschformat los. Mittlerweile existiert z. B. eine für Webservices einfach nutzbare [JSON](#)-basierte Serialisierung für Linked Data, s. [SLK⁺14].

⁵⁸Für diese Szenario ist es hilfreich, für die Identifizierung von Aussagegegenständen stets mehrere Identifier anzugeben und diese damit als äquivalente Bezeichner zu deklarieren.

Neben [RDF](#)-basierten oder auf Tupeln beruhenden Herangehensweisen wurden weitere alternative Semantikmodelle entwickelt. Mit Topic Maps [\[ISO13250\]](#) wurde ein ISO-Standard geschaffen, der sich u. a. in das *Topic Maps Reference Model* mit grundlegenden Aussagen zu Identität von Entitäten und das *Topic Maps Data Model* mit konkreten Modellierungskonstrukten gliedert. Das darin beschriebene Assoziationsmodell erlaubt es, in einer einzigen Aussage mehr als zwei Aussagegegenstände in Beziehung zu bringen, wobei es möglich ist, die jeweils von ihnen eingenommene Rolle genauer zu spezifizieren. Im praktischen Einsatz konnte sich die Topic-Maps-Technologie jedoch nicht durchsetzen.

Trotz komplett unterschiedlicher Ausrichtung, divergierender Auffassung zur Festlegung eines Datenschemas und stark getrennter Werkzeug-Umgebungen herrscht insgesamt doch eine große Verwandtschaft zwischen Semantic-Web-Wissensbasen und Graphdatenbanken. Diese zeigt sich u. a. in den Abfragemöglichkeiten, die im nächsten Abschnitt vorgestellt werden sollen.

2.3.6 Abfragesprachen

Wie in vielen Programmier- und Abfragesprachen wird auch im Umfeld der Graphdatenbankabfrage in deklarative und imperative Sprachen (bzw. Sprachteile) unterschieden. Deklarativ wird notiert, welche Eigenschaften und Strukturen in den Ergebnissen gewünscht sind, woraufhin eine Ausführungs-Engine die entsprechend dafür notwendigen Berechnungsschritte selbstständig ermittelt und in selbst festgelegter Sequenz durchführt. Imperativ werden vom Nutzer einzelne Abarbeitungsschritte festgelegt, die dann in vorgegebener Reihenfolge ausgeführt werden.

Die Abfrage von Graphdatenbanken ist aus wissenschaftlicher Sicht gut erforscht. Umfangreiche Vergleiche und eigene Weiterentwicklungen von Abfrageansätzen werden z. B. in [\[BLLW10\]](#) und [\[Woo12\]](#) vorgenommen. Für das bereits vorgestellte Tupelsystem „graphd“ wird in [\[MDGM10\]](#) eine Abfragesprache präsentiert, die einen sehr niedrigen Abstraktionsgrad gegenüber der genutzten Speicherstruktur aufweist. In Freebase wurde aufbauend darauf eine [JSON](#)-basierte Abfragesprache eingeführt, die Abfragen auf der Ebene des logischen Datenbankschemas ermöglicht, vgl. [\[Fla09\]](#). Dabei handelt es sich um eine deklarative mit *Templates* arbeitende Sprache – es werden dort „Vorlagen“ für die Ergebnisausgabe erstellt, deren Lücken dann durch Bindung mit passenden realen Daten der Abfrageergebnisse gefüllt werden. Grundsätzlich damit vergleichbar ist die von Facebook

vorgestellte (auf die Anfrage an Facebook-APIs beschränkte) Sprache GraphQL⁵⁹.

Für die Mustersuche in Graphen wurden in den vergangenen Jahren viele theoretische und technologische Fortschritte erzielt, s. z. B. [FLM⁺10]. Ein effizientes *Pattern Matching* bildet auch die Grundlage von *SPARQL Protocol And RDF Query Language (SPARQL)*, einer deklarativen Abfragesprache, die hauptsächlich im Umfeld des Semantic Web eingesetzt wird, z. T. aber auch von Property-Graph-Systemen unterstützt wird. Für die im deklarativen Bereich notwendige Ausführungsoptimierung für Anfragen ist die grundsätzlich „schemafreie“ Natur von RDF-Graphen wenig geeignet. Werden die in RDF kodierten Schemainformationen von der Datenbank nicht gesondert behandelt, kann ein einfaches Tripel zwei verschiedene Klassen oder Eigenschaften gleichsetzen und damit Millionen von Instanzdatensätzen beeinflussen, so dass die komplette Topologie des Graphen sich ändert. Die für eine automatische Query-Optimierung nötige statistische Erfassung von Häufigkeiten bezüglich Klassen, Eigenschaften und Eigenschaftswerten kann in einem so freien Umfeld nicht effizient erfolgen.

Bei Property-Graph-Datenbanken ist das Schema zwar ebenfalls einfach anpassbar, wird jedoch nicht in Form von Datenbankeinträgen verwaltet. Auch sind im laufenden Betrieb keine willkürlichen Änderungen an bestehenden (und mit Indizes versehenen) Eigenschaften oder Kanten typen möglich. Das ermöglicht es deklarativen Abfragesprachen, wie „Cypher“⁶⁰, einen Ausführungsplan statistisch informiert aufzustellen. Über das Projekt openCypher⁶¹ wird diese Abfragesprache derzeit auch für den Einsatz außerhalb von Graphdatenbanken erweitert.

Auf der Seite der Abfragesprachen mit imperativem Anteil besitzt Gremlin, s. [Rod15], die größte Verbreitung. Bei Gremlin-Abfragen wird üblicherweise eine Kette von Traversierungsoperationen beschrieben, nach deren Ausführung die gesuchten Elemente selektiert sind. Bei der Planung effizienter Abfragen wird Modell- und Domänenwissen benötigt. Im Gegenzug fällt eine zielgerichtete Optimierung leichter, da innerhalb der Abfragen alle Details der Ausführung einzeln gesteuert werden können. Gremlin ist Bestandteil von Apache TinkerPop und kann daher als Basis genutzt werden, um mit der gleichen Abfragesprache sowohl OLTP-Anfragen (in Graphdatenbanken im eigentlichen Sinne dieser Arbeit) und OLAP-Anfragen (in sogenannten *Graph Processors*) zu formulieren.

⁵⁹<http://graphql.org/learn/>

⁶⁰<http://neo4j.com/developer/cypher-query-language/>

⁶¹<http://www.opencypher.org/>

Andere Daten- und Semantikmodelle bringen jeweils ihre eigenen Abfragemethoden mit sich. Für Topic Maps existiert unter anderem die [Topic Maps Query Language \(TMQL\)](#). Die Abfrage von Baumstrukturen z. B. über Pfadausdrücke kann in [XML](#)-Datenbanken über Werkzeuge zur Elementselektion der [XSL](#) erfolgen. Letztlich sind auch Abfragesprachen für relationale Datenbanksysteme (allen voran [SQL](#)) zur Graphabfrage geeignet, solange zur Bestimmung des Ergebnisses nur überschaubar viele *Join-Statements* notwendig sind.

2.4 Vorarbeiten und verwandte Gebiete

Nach der ausführlichen Vorstellung der drei großen Themenkomplexe aus dem Titel dieses Buches sollen nun Themen herausgegriffen werden, die eine besondere inhaltliche oder technologische Nähe zu den in dieser Arbeit vorgestellten Überlegungen und Ergebnissen besitzen.

Eine umfangreiche und vielschichtige Betrachtung zum Wesen von Text und zu den bisherigen Möglichkeiten der Textrepräsentation wird in [\[Sah13\]](#) aus Sicht der wissenschaftlichen und praktischen Editorik stellvertretend für viele textbezogene geisteswissenschaftliche Fachrichtungen vorgenommen. Es ist praktisch nicht möglich (und darüber hinaus wohl auch nicht sinnvoll), für solche umfassenden Ansätze der Systematisierung mit all ihren vielschichtigen Sichtweisen allgemeine digitale Repräsentationsformen zu finden. Im Folgenden werden daher Ansätze und Werkzeuge vorgestellt, die Teilaspekte der Textrepräsentation, Textanalyse und Textrecherche im Rahmen der e-Humanities bearbeiten.

Zum einen existieren zahlreiche generische Text-Mining-Werkzeuge, allen voran die Voyant Tools⁶² von Sinclair, Stéfán und Rockwell. Diese werden auf der Webseite als „*web-based reading and analysis environment for digital texts*“ vorgestellt. Sie bieten dem Nutzer einen schnellen Einstieg in digitale Textanalyse, u. a. auch, weil bei webbasierten Systemen keine Hürden für eine Softwareinstallation existieren und weil einige kleine Beispielkorpora bereits im System angeboten werden und zum Einarbeiten in das Werkzeug einladen. Die Voyant Tools vereinen etablierte Visualisierungsansätze mit einfacher frequenzbasierter Wortstatistik, berücksichtigen jedoch für die Analysen keine Metadaten, Strukturinformationen oder Annotationen. Der Funktionsumfang der Voyant-Tools (und ihrer Vorgängersoftware „HyperPo“) werden in [\[Joc13\]](#) als „*self-serve analysis tools for*

⁶²<http://voyant-tools.org/>

traditional concordance and co-occurrence alongside more experimental widgets for the processing and deforming of textual data“ beschrieben. Viele der experimentellen Funktionen werden jedoch nicht länger angeboten, sind nicht gewartet oder nicht in neuere Versionen übertragen worden. Aus der Entwicklung der Werkzeuge sind jedoch viele interessante Erkenntnisse entstanden, die durch entsprechende Publikationen verbreitet werden. Sinclair gibt z. B. in [Sin03] den wichtigen Denkanstoß: „*Design of new tools [...] should give full space to how literary critics interact with texts, rather than simply focus on what computers can do well.*“

Daneben gibt es noch eine große Zahl spezialisierter Rechercheumgebungen, welche an einzelne Projekte oder Korpora gebunden sind oder welche für bestimmte Einsatzzwecke in Forschungsinfrastrukturen vorgehalten werden. Erstere sind z. T. als *Open Source Software* frei verfügbar und so (zumindest theoretisch) auf eigene Bedürfnisse anpassbar. Letztere Systeme lassen sich durch die Serviceorientierung der Infrastrukturen meist mit weiteren Werkzeugen kombinieren, dagegen aber in der Regel nicht umfassend anpassen.

Die Erarbeitung eines Überblicks über alle existierenden Werkzeuge allein böte genügend Stoff für eine eigenständige Dissertationsschrift, weshalb an dieser Stelle nur auf wesentliche Vorarbeiten im Sinne der vorgestellten Forschungsfragen eingegangen werden kann. Für die Verbindung von Textmodell und Graphenrepräsentation existieren einige Präzedenzfälle, die hier eine eingehendere Erwähnung finden sollen:

Seit 2006 existiert ein jährlich im Rahmen verschiedener Konferenzen abgehaltener „*Workshop on Graph-based Algorithms for Natural Language Processing*“⁶³. Dort liegt der Fokus auf Algorithmik und dem Aufbau einer Graphstruktur als Vorverarbeitungsschritt für Verfahren der automatischen Sprachverarbeitung. Auch im Rahmen anderer Veranstaltungen und in Journalen wird diese Arbeitsweise zuweilen aufgegriffen.

Es kommen dabei z. T. sehr unterschiedliche Formen der Graphinduktion zum Einsatz, da die Entscheidung, welche Analyseeinheiten durch Knoten und Kanten repräsentiert werden sollen, in diesen Szenarien stark vom anzuwendenden Algorithmus (und natürlich der Zielstellung) abhängt.

In [MT04], wo die Anwendung des PageRank-Algorithmus⁶⁴ auf graphförmige Textrepräsentationen vorgestellt wird, heißt es zu dieser Thematik:

⁶³<http://www.textgraphs.org/>

⁶⁴s. [PBMW99]

„Depending on the application at hand, text units of various sizes and characteristics can be added as vertices in the graph, e.g. words, collocations, entire sentences, or others. Similarly, it is the application that dictates the type of relations that are used to draw connections between any two such vertices, e.g. lexical or semantic relations, contextual overlap, etc.“

Oft wird dabei eine sehr kondensierende Sichtweise gewählt, die auf bestimmte Anwendungsfälle abgestimmt ist, wie z. B. in [RV13], wo ein „Graph-of-word“-Modell als ungewichtetes gerichtetes Netzwerk von Kookkurrenztermen aus der Sequenz des Textes abgeleitet wird.

Ansätze aus der Linguistik modellieren Texte meist feingliedriger, etwa unter Angabe des Abhängigkeitsgraphen, Abbildung einzelner Wörter des Fließtexts, sequenzieller Verknüpfung von Sätzen, aber auch weiteren relevanten Aspekten, wie einer verknüpften Repräsentation von Koreferenz, vgl. [MES07]. Die Graphenrepräsentationen verschiedener gängiger Annotationsformate können in einer gemeinsamen Graphstruktur zusammengeführt und vereinheitlicht werden, wie etwa in [IS07] vorgestellt. Seit diesen Veröffentlichungen von 2007 ist jedoch keine universelle Adaption dieser Ideen innerhalb der Computerlinguistik oder in die Korpuslinguistik hinein zu beobachten.

Neuere Arbeiten zur nicht-persistenten Repräsentation von Korpora mit mehrschichtigen linguistischen Annotationen umfassen z. B. die Arbeiten von Neumann, s. [Neu15], für die verlustlose Konvertierung zwischen linguistischen Annotationsformaten. Eng verwandt damit sind die Arbeiten zu „Salt“⁶⁵, einem „Theorie-neutralen Metamodell“ für linguistische Annotationen, s. [ZR10].

Ansätze, in denen der Graph nicht nur als Speicherungsform oder Werkzeug zur Formatkonvertierung angesehen wird, sondern als umfassender digitaler Repräsentant des textuellen Untersuchungsgegenstandes, kommen insbesondere bei Textanalysen auf lexikalisch-semantischer Ebene zum Einsatz. In [MGWD10] werden z. B. die Metadaten der Dokumente verwendet, um die korpusweite zeitliche Dynamik der Interaktion von Vokabular (in diesem Fall auf der Ebene von Lexemen) zu untersuchen. Der Fokus liegt dabei auf einer Nutzung linguistischen Wissens für die Analyse der Texte, wobei die Ideen für die Netzwerkinduktion von kookkurrenzbasierten Modellen bis zur Nutzung von *Word Embeddings* weiterentwickelt wurden, vgl. [EM16]. Einen dazu passenden (nicht auf Graphentechnologien basierenden) Beitrag zur feingranularen Aggregation von

⁶⁵<http://corpus-tools.org/salt/>

Kollokationen (also Kookkurrenzwörtern in gewünschten Kontexten) in großen Korpora auf der Basis von Dokumentzeitstempeln bietet das Werkzeug DiaCollo, s. [JGW16].

In einigen allgemeineren Ansätzen ohne Fokus auf das *Text Mining* wird versucht, das abstrakte Modell der TEI-Spezifikation von XML losgelöst zu betrachten und z. B. mit Semantic-Web-Methoden zu modellieren, wie etwa in [TMK⁺06] und [CSFF16]. Schmidt fasst in [Sch10] umfassend die bestehenden Bedenken gegen das OHCO-Modell zusammen und schlägt als Nachfolgetechnologie sogenannte „Multi-Version Documents“ vor, die in einer Graphrepräsentation vorliegen und mehrere Varianten und Lesarten gleichzeitig abbilden können. Anwendungsfälle einer solchen Kodierung von Textvarianten in der Manuskripterfassung wurden dabei bereits in [Sch06] vorgestellt.

Nicht nur Texte, sondern auch lexikalische Ressourcen, die für deren Analyse herangezogen werden können, lassen sich gewinnbringend in Graphenform vorhalten und darin zielgerichtet abfragen. Als konzeptuelle „Weiterentwicklung“ digitaler Thesauri wurde bereits Mitte der 1980er Jahre, mit WordNet⁶⁶ eine als Netzwerk strukturierte Ressource geschaffen, in der Beziehungen nicht mehr nur auf der Ebene von Wörtern, sondern von Wortbedeutungs-Einheiten⁶⁷ modelliert sind, s. z. B. [Fel98]. Seitdem wurde z. B. in [Tri06] mit dem *Lexicon Graph* eine graphbasierte Lösung zur Zusammenführung verschiedener lexikalischer Ressourcen geschaffen und mit UBY⁶⁸ ein sehr umfassende Datensammlung erstellt, s. [GEKH⁺12], welche sich über einen RDF-Export in eine Graphdatenbank einlesen lässt, vgl. [EKMC15]. Eine gleichzeitige Betrachtung von lexikalischen Ressourcen und Textannotationen ermöglicht die Arbeit von Damerow zu einer Forschungsumgebung, die auf die „Meso-Skala“, also den Bereich zwischen Mikro- und Makrosicht auf Text, abzielt, s. [Dam14]. Dort wird eine Graphstruktur aus Quadrupeln⁶⁹ modelliert.

Neben text- und wortbezogenen Modellen werden auch solche Ansätze verfolgt, die sich bewusst auf Aspekte der Strukturrepräsentation beschränken. Im Jahr 2008 wurde von Liu und Smith in Reaktion auf die Unzulänglichkeiten des OHCO für die digitale Erschließung von Manuskripten auf Grundlage des relationalen Datenbankmodells in [LS08] eine flexible Modellierung einzelner Text enthaltender Objekte unterschiedlicher Granularität vorgestellt. Diese werden in Sequenz und als Netzwerk, welches aus unterschiedlichen „Kantentypen“ aufgebaut ist, beschrieben. Die einzelnen „Sprachobjekte“ werden dabei allerdings nicht genauer charakterisiert. Dazu heißt es:

⁶⁶<http://wordnet.princeton.edu/>

⁶⁷sogenannten *Synsets*, welche auf Synonymie basieren

⁶⁸<http://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/>

⁶⁹welche als „kontextualisierte“ Tripel verstanden werden

An alternate model might define text, therefore, as "a labelled network of language objects." A "language object" is like the "content object" in the OHCO model; examples of language objects are chapters, lines, pages, paragraphs, words, [...]

This is a more flexible model than the hierarchical, where the relationships between elements are implied by the hierarchical structure [...]

The network model is more useful for exploratory projects [...]

Ein Projekt mit ähnlichem Fokus auf die Modellierung von Struktur allein ist die bereits erwähnte CITE-Infrastruktur mit den CTS-Zugriffsmöglichkeiten. Dort wird für kanonischen Text, also Dokumente oder ganze Korpora mit jeweils wohl definierten Primärhierarchien für die Strukturierung eine technische Repräsentationsform über Semantic-Web-Technologien geschaffen, die als Spiegel der traditionellen Editionspraxis fungieren soll.

Aus den Reihen der Digitalen Geisteswissenschaften werden derzeit darüber hinaus einige Experimente zur Nutzung von Graphdatenbanken durchgeführt. In [KA16] liegt dabei der Schwerpunkt in der Stemmatalogie, der netzwerkartigen Auflistung von handschriftlich niedergeschriebenen Varianten eines Textes zur Rekonstruktion seiner Überlieferungsgeschichte. Auch dort müssen dazu feingliedrigere Modellierungsebenen der einzelnen Texte beachtet und abgebildet werden. Dabei wird der Text als dokumentübergreifend verknüpfte Sequenz von Textteilen in der Graphdatenbank abgebildet, ohne dass jedoch ein graphbasiertes Modell für diese Textteile selbst entwickelt wird. Es heißt dazu: „So sind Einheiten, wie zum Beispiel Single-Wörter, Sätze oder ganze Paragraphen, etc. vorstellbar.“

Aus einer Reihe weiterer Experimente zur Nutzung von Graphdatenbanken in den Geisteswissenschaften⁷⁰ ist ein kürzlich erschienener Beitrag im Bereich der digitalen Editionswissenschaft hervorzuheben: In [Kuc16] wird eine erste, sehr vielversprechende Machbarkeitsstudie zur Abbildung von TEI-Editionen in Property-Graph-Datenbanken vorgestellt. Auch hier wird jedoch kein besonderes Augenmerk auf eine Weiterverarbeitung der Daten mittels *Text-Mining*-Verfahren oder die Abbildung großer Korpora gelegt.

⁷⁰s. <http://mittelalter.hypotheses.org/5089> und <http://mittelalter.hypotheses.org/5995>

2.5 Ableitbare Systemanforderungen

Unter Berücksichtigung der bis hierhin skizzierten Bedürfnisse digitaler geisteswissenschaftlicher Forschung und der aktuellen Entwicklung im technologischen Umfeld der Textrepräsentation, lassen sich einige zentrale Anforderungen an ein neuartiges, ergänzendes Rechtersystem definieren.

Das Ziel, in dieser Arbeit die Grundlage für alternative Ansätze bei der Entwicklung textbezogener Rechtersysteme zu schaffen, geht mit einer technologischen Abkoppelung von bisherigen Standardlösungen einher. Bei der Konzeption dieser Alternative sollen die Stärken der verschiedenen oben gelisteten Vorarbeiten berücksichtigt werden. Da die Entwicklung von Rechtersystemen stets ein interdisziplinärer Prozess ist (oder zumindest sein sollte), sind – auf verschiedenen Ebenen – auch unterschiedliche Nutzergruppen involviert. Die Systemanforderungen aus Sicht von Informatikern, Programmierern und Nutzern aus den Sozial- und Geisteswissenschaften sind divers und sicher nicht in jedem Fall miteinander vereinbar. Die Anwendungsgebiete sind darüber hinaus deutlich zu breit für die Durchführung einer verallgemeinernden softwaretechnischen Domänenanalyse, z. B. entsprechend [KCH⁺90]. Die letztendliche Spezifikation eines spezialisierteren Rechtersystems muss daher einzelfallbezogen in einem gemeinsamen Prozess enger Abstimmung stattfinden, der im Rahmen dieses Buches so nicht vollzogen werden kann. Dem Autor ist bewusst, dass die Auswahl und Wichtung der folgenden Aspekte daher durchaus subjektive Züge trägt.

Es wurde jedoch darauf geachtet, die Anforderungen so zu formulieren, dass sie nicht einer konkreten finalen Spezifikation vorweggreifen, sondern vielmehr deren Findung unterstützen. Über den vorgestellten Aspekten steht eine zentrale Kernanforderung: die Schaffung einer flexiblen Technologie. Flexibilität ist dabei eine schwer zu definierende Größe. Sie wird in [NYSC00] als „*measure of the potential rather than performance*“, „*user- or situation specific*“ und zudem „*difficult to determine a priori*“ beschrieben. Nur wenn die erstellte Software durch Erweiterungsmöglichkeiten oder generische Lösungsansätze auch außerhalb ihrer ursprünglichen Spezifikation einsetzbar ist, kann auf die Fülle an Themen und Forschungsperspektiven sowie die Besonderheiten von Quellensammlungen, welche die Anwendungsdomäne mit sich bringt, adäquat reagiert werden. Unter Berücksichtigung dieser angestrebten systematischen Flexibilität und mit Blick auf die bestehenden Werkzeuge und vorgestellten Vorarbeiten lassen sich im Einzelnen die folgenden Schlagwörter herausgreifen:

Non-destruktive Datenhaltung:

Die Akzeptanz eines Recherchesystems steht und fällt mit dem Vertrauen in seine Quellentreue. Falls die ins System eingespeisten Roh-Textdaten für weitergehende Analysen umgeformt werden müssen, (z. B. um nicht-standardisierte Orthographie zu vereinheitlichen), so muss jederzeit die Möglichkeit zur Rekonstruktion der Originaltexte aus der internen Repräsentation heraus bestehen. Dabei sollte der Rückschluss von den transformierten auf die originalen Stellen möglichst direkt erfolgen können. Der Nutzer sollte auch nach dem initialen Einlesen die Möglichkeit zur forschungsgeleiteten Anpassung der internen Repräsentation haben, ohne das Original separat verändern zu müssen oder seine Integrität unabsichtlich zu kompromittieren.

Vorberechnungsfreie Statistiken:

Da zu erwarten ist, dass sich in explorativen Untersuchungen der Textbasis häufig der Analyseschwerpunkt ändert, und viele unterstützende statistische Auswertungen dann am aussagekräftigsten sind, wenn Sie direkt den aktuellen Kontext betreffen, müssen Möglichkeiten geschaffen werden, statistische Berechnungen so vorzunehmen, dass sie diesen häufigen Kontextwechseln in besonderem Maße gerecht werden. Statt, wie üblich, zunächst das manuelle Definieren von Subkorpora zu erfordern, um anschließend eine *Offline*-Neuberechnung von Statistiken auf Basis dieser eingeschränkten Textsammlung vorzunehmen, sollten statistische Auswertungen möglichst ohne Vorberechnung und direkt auf der bestehenden Datenbasis unter Einschränkung auf den aktuellen Kontext durchführbar sein.

Interaktive Bedienung mit vertretbaren Antwortzeiten:

Die Steuerung des Analysevorgangs durch den Forscher muss zu jedem Zeitpunkt eine Änderung von Parametern und Rückführung interessanter Resultate auf neue Recherchen ermöglichen. Lange Wartezeiten zwischen dem Absetzen einer Anfrage und der Anzeige von Ergebnissen hemmen dabei die Produktivität und stören überdies den immersiven Recherchevorgang. Bei potentiell langen Vorgängen ist eine interaktive Rückmeldung über den aktuellen Bearbeitungsstand wichtig. Zudem ist es wünschenswert, etwaige Zwischenergebnisse ohne unnötige Wartezeiten bis zum Abschluss des Gesamtvorgangs zu erhalten. Das Abbrechen von nicht länger gewünschten Anfragevorgängen soll jederzeit möglich sein.

Unterstützung navigierender und explorierender Visualisierungstechniken:

Zusätzlich zu interaktiven und vorberechnungsfreien Abfragen muss ein exploratives Recherchesystem in der Lage sein, eine lokale Sicht auf die Daten abzubilden. Angren-

zende Kontexte müssen direkt erreichbar und einfach aggregierbar sein. Eine Erhöhung und Verringerung des Detailgrads ausgegebener Informationen sollte möglich sein, um Abstraktion und das Aufdecken von Abhängigkeiten zu ermöglichen.

Gute Integrierbarkeit bei Vermeidung technologischer „Medienbrüche“:

Das System soll flexible Möglichkeiten für den Import von Daten aus Quellensammlungen und Drittsystemen bieten. Ein großer Möglichkeitsraum für Transformation und Abfrage soll die Notwendigkeit für einen gleichzeitigen Einsatz weiterer Analysesysteme reduzieren, um eine schwer zu synchronisierende und potentiell inkonsistente Datenhaltung in mehreren Systemen zu vermeiden. Um jedoch kein „Datensilo“ zu erzeugen, soll das System ebenso flexible Möglichkeiten für die Serialisierung und den Export für die Weiterverarbeitung von Daten und Rechercheartefakten in anderen Programmen zur Verfügung stellen und unabhängig von konkreten Austauschformaten sein.

Weitestgehende Sprach- und Skriptunabhängigkeit:

Für eine breite Anwendbarkeit des Systems muss gewährleistet sein, dass es mit einer Vielzahl an Quellensorten und Quellensammlungen kompatibel ist. Insbesondere sollten keine Beschränkungen bei alternativen Schreibrichtungen (u.a. von rechts nach links), ungewöhnlichen Alphabeten oder Texten, die mit Sonderzeichen durchsetzt sind, auftreten. Die Übertragbarkeit für die damit entwickelten Verfahren sollte durch die Entkopplung von sprachabhängigen und sprachunabhängigen Aspekten im Datenmodell und im Basisprogramm begünstigt werden. Bei all dieser methodischen Offenheit sollte jedoch berücksichtigt werden, dass keine voreilige Ausrichtung auf seltene Sonderfälle geschieht: Standardfälle sollten sich sehr einfach handhaben lassen und komplexe Randphänomene lediglich grundsätzlich (ggf. mit angemessenen Zusatzaufwänden) abbildbar sein.

Kapitel 3

Kadmos – ein graphbasiertes Recherchesystem

Οἱ δὲ Φοίνικες οὗτοι οἱ σὺν Κάδμῳ ἀπικόμενοι, τῶν ἦσαν οἱ
Γεφυραῖοι, ἄλλα τε πολλὰ οἰκήσαντες ταύτην τὴν χώραν ἐσήγαγον
διδασκάλια ἐς τοὺς Ἕλληνας καὶ δὴ καὶ γράμματα, οὐκ ἔόντα πρὶν
Ἑλλήσι ὥς ἐμοὶ δοκέειν, πρῶτα μὲν τοῖσι καὶ ἅπαντες χρέωνται
Φοίνικες.

*Diese Phönizier, die mit Kadmos kamen, unter denen sich auch die
Gephyräer befanden, haben mit sich viele Lehren zu den Hellenen
gebracht, insbesondere auch die Schriftzeichen – welche diese vorher
nicht hatten, wie ich meine – die anfangs gleich den phönizischen
waren.*

Herodot von Halikarnassos
Griechischer Historiker und Geograph

[Hdt. 5.58.1], ins Deutsche, entsprechend des [HERODOT-Korpus](#)

3.1 Entwicklungsziele

Bei der Entwicklung der Rechercheanwendung Kadmos¹ wird das Ziel verfolgt, die im letzten Kapitel beschriebenen Technologien so einzusetzen, anzupassen und zu erweitern, dass die herausgearbeiteten Systemanforderungen erfüllt sind. Konkret bedeutet das, dass ein System geschaffen werden muss, durch welches eine breite Anwendbarkeit der Technologie für eine Vielzahl denkbarer Anwendungsfälle gewährleistet ist. Entsprechend sollte diese gewünschte Flexibilität anhand verschiedener Korpora in verschiedenen Alphabeten und Sprachen nachgewiesen werden.

Da das System quantitative Aussagen für den Forschungsbereich ermöglichen soll, muss die Korrektheit zurückgelieferter Ergebnisse gewährleistet sein, so dass keine Approximationen im Kern des Systems stattfinden dürfen. Sind (potentiell schneller zu erlangende) Näherungen für einen Anwendungsfall zulässig, so sollten diese explizit vom Nutzer angefordert werden. Das System muss dabei genügende Anpassbarkeit aufweisen, um dafür gegebenenfalls nötige Hilfskonstrukte in das Datenmodell einfügen zu können bzw. die entsprechenden Abfragen in einer alternativen Art und Weise zu formulieren, so dass näherungsweise Resultate zurückgegeben werden können. Insgesamt soll das Grundsystem so wenige Annahmen über die abzubildenden Quellen und die späteren Anfrageszenarien tätigen, wie möglich.

Im Sinne universeller Nutzbarkeit und guter Wartbarkeit soll das System seine Funktionen über serverbasierte Dienste anbieten, um eine lose Kopplung der Komponenten zu erreichen. Werden die Services dann über webbasierte Schnittstellen angeboten, erlaubt dies eine flexible ortsungebundene Nutzung auch auf leistungsschwächeren Endgeräten.

Im Lichte allgemein steigender Anforderungen an serverbasierte Softwaresysteme, etwa im Hinblick auf kurze Antwortzeiten, massiv-parallele Zugriffe, ständige Erreichbarkeit und nur durch Hardwaregrenzen limitierte Erweiterbarkeit des Datenbestandes, haben sich neue Entwurfsmuster für komplexe Anwendungen entwickelt. Der Aufbau von Kadmos soll sich an den Architektur- und Entwicklungsrichtlinien des 2014 konzipierten *Reactive Manifesto*² orientieren, insoweit das für eine prototypische akademische Software (mit perspektivisch zunächst eng umgrenzter Nutzerzahl) sinnvoll ist.

¹Der Name Kadmos leitet sich nicht von einem technischen Akronym ab, sondern ist von der gleichnamigen Sagengestalt des klassischen Altertums entlehnt, die hier symbolisch für den Technologietransfer im Bereich verschriftlichter Sprache stehen soll.

²<http://www.reactivemaneifesto.org/>

Konkret bedeutet das zum einen, dass die angesprochene lose Kopplung von Programmteilen (wie sie in modularen Softwaresystemen und objektorientierten Umgebungen bereits lange praktiziert wird) wo immer es möglich und sinnvoll ist, durch asynchrone Anfrage-Ausführung und Kommunikation der Komponenten unterstützt werden soll. Dadurch werden Latenzen verringert und Ressourcen effektiver genutzt. Weiterhin soll eine Skalierung des Systems durch *Sharding* und *Replikation* auch auf Cluster aus mehreren Rechnern unterstützt werden. Dabei sollen Daten persistent gespeichert und in konsistenter Form abgefragt werden können. Das Antwortverhalten für übliche Abfragen soll sich zudem im Bereich der *Realtime*-Interaktion, also in der Regel maximal im Bereich von Sekunden, nicht von Minuten bewegen.

Bei der Entwicklung von Kadmos soll zudem ein besonderes Augenmerk auf prototypische Entwicklung gelegt werden. In [GR10] argumentieren Galey und Ruecker, dass Prototypen, da sie so viel implizites Wissen, Abstraktionen und Vorannahmen enthalten, genau wie wissenschaftliche Papiere (und zunehmend auch digitale Datensammlungen) einem *Peer-Review*-Prozess unterzogen werden sollten. Für eine solche zukünftige Anforderung ist es von Vorteil, über einheitliche Modellierungs- und Ausführungsumgebungen zu verfügen. Hat das Kernsystem erst einmal im Review-Prozess bestanden, muss ggf. künftig nur noch die jeweilige Erweiterung von Experten untersucht werden.

Ruecker unterscheidet in [Rue15] für das Umfeld von Digital-Humanities-Projekten drei Arten der prototypischen Systementwicklung: Bei „produktionsgetriebenen“ Prototypen steht die schrittweise Verfeinerung eines Rohprodukts hin zu einem stabil lauffähigen Produktivsystem im Fokus. Bei „experimentellen“ Prototypen ist das Ziel eher, generalisierbares Wissen über Daten, Methodik und Forschungsfragen aus dem Erstellungsprozess abzuleiten. Schließlich werden bei so genannten „*provotypes*“ auf provokative Weise Gewohnheiten und Erwartungen durch innovative Herangehensweisen hinterfragt und herausgefordert, was sich meist in alternativen Nutzeroberflächen und Interaktionsformen widerspiegelt.

Mit Kadmos soll eine geeignete Plattform für alle drei dieser Herangehensweisen geschaffen werden. Ziel ist es, eine robuste technologische Grundlage für neuartige Rechewerkzeuge über ein feingliedriges, viele Kontexte zugänglich machendes Datenbanksystem mit interaktiven Abfragemöglichkeiten und großem Erweiterungspotential zu schaffen. Das dafür verwendete digitale Ordnungs- und Zugriffsschema wird im nächsten Abschnitt vorgestellt.

3.2 Daten- und Domänenmodell

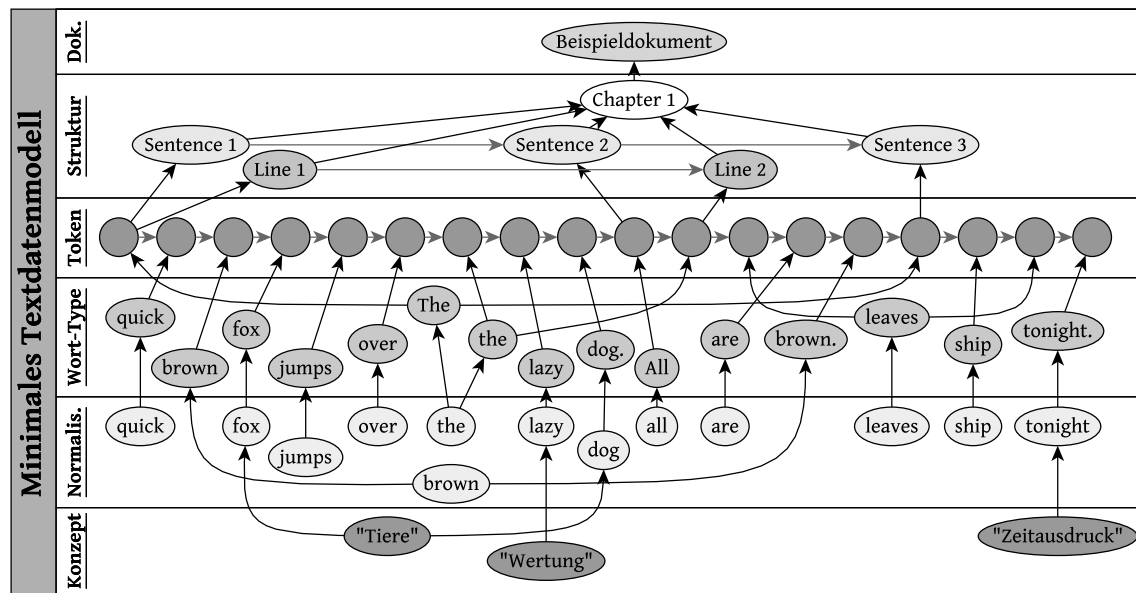
Für die Modellierung von Korpora, aber auch von Querschnittsaspekten, wie der Nutzerdaten- und Rechteverwaltung, wird in Kadmos das Property-Graph-Modell verwendet. Falls dabei für Knoten grundsätzlich mehrere Werte für eine Eigenschaft zugelassen werden sollen (z. B. Namen von Autoren, Login-Identitäten von Nutzern), so werden die entsprechenden Eigenschaften-Wert-Paare jeweils in einem neuen Knoten abgespeichert, welcher über eine entsprechende Kante mit dem Original verbunden wird. So wird es möglich, jeden der zugewiesenen Werte einzeln zu adressieren, und bei Bedarf z. B. in einen Namensknoten eine Property zum Namenssytyp (mit Werten, wie „Geburtsname“ oder „Krönungsname“) zu notieren oder der verknüpfenden Benennungs-Kante eine Eigenschaft „bevorzugt als Anzeigename verwenden“ beizufügen.

Eigenschaftenwerte in Zeichenkettenform werden dabei durch ein externes Indexsystem verwaltet, wobei die Synchronisierung mit der Datenbasis automatisch geschieht. In der Theorie kann eine solche Indizierung auch direkt über Konstrukte der GraphDB geschehen, wie z. B. [RN11] zeigt. Für die Anwendung in Kadmos ist jedoch die Separierung dieser Aspekte in zwei spezialisierte und korrespondierende Systeme absolut zweckmäßig.

Im interaktiven prototypischen Entwicklungsszenario ist die Introspektionsfähigkeit für momentan selektierte Objekte ein wichtiges Arbeitsinstrument: Einzelne Elemente des Graphen sollten auch bei Abfragen mit reduziertem Kontext noch aussagekräftige Konstrukte sein. Da das Property-Graph-Modell zwar Kantenlabels kennt, aber Knoten keinen Typ besitzen, wird dieser in Kadmos für alle Knoten über eine Property (namens `node_type`) emuliert. Aus Effizienzgründen wird dabei ausnahmsweise bei Knoten vom Typ Token (als am häufigsten auftretendem Knotentyp) auf die Speicherung einer solchen Property verzichtet. Dadurch ist das Nicht-Vorhandensein der Property ein genauso eindeutiges Knotentyp-Merkmal wie es ein entsprechender Propertywert wäre. Die Werte werden platzsparend in numerischer Form gespeichert, sind im Programm aber über ihre Namen nutzbar.

Die Grundzüge des im Folgenden beschriebenen Datenmodells für die Repräsentation von Text wurden bereits in [Efe15] vorgestellt. Sie ergeben sich aus der intuitiven Abbildung von Textelementen, deren Hierarchie und Sequenz sowie weiteren für die Recherche notwendigen Angaben in Graphenform. Anders als etwa in [Kuc16] wird in Kadmos dabei in Types und Tokens unterschieden. Zum einen verbessert dies die Ausnutzung von Spei-

cherplatz, indem Redundanzen vermieden werden. Zum anderen ergibt sich darüber eine wichtige Traversierungsachse für das *Text Mining*, wie später noch demonstriert wird. Abbildung 3.1 zeigt (unter Ausblendung von Kantentypen) die wesentlichen Modellkonstrukte, die für die Repräsentation eines Dokuments zum Einsatz kommen.



The quick brown fox jumps over the lazy dog. All the leaves are brown. The ship leaves tonight.

Abbildung 3.1: Schematische Darstellung der Instanzdatensätze und Verknüpfungen eines kurzen Beispieldokuments bei minimalistischem Textdatenmodell

Die Unterteilung des Dokuments in Token erfolgt in diesem Beispiel an Leerstellen. Allgemein wird ein solches einfaches und relativ sprachunabhängiges Tokenisierungsmodell bevorzugt, in denen nicht-freistehende Satz- und Sonderzeichen nicht als Einzeltoken betrachtet werden. Solche Sonderzeichen innerhalb der verknüpften Types werden bei der Normalisierung eliminiert. Die „bereinigten“ Versionen der Types werden als „normalisierte Types“ abgespeichert und mit den entsprechenden Types verknüpft. Dieses Verfahren wird noch in Abschnitt 3.6 auf Seite 91 detailliert vorgestellt.

Die Token sind die zentralen Elemente des Graphen, sie selbst besitzen standardmäßig jedoch keine Eigenschaften. Ihre ganze Bedeutung ergibt sich aus Ihrem Kontext, welcher über Kanten zu anderen Knoten hergestellt wird. Abbildung 3.2 auf der nächsten Seite zeigt die verschiedenen Pfade, die vom Token ausgehend zum einen über Strukturelemente in Richtung der Dokumente und Metadaten existieren, als auch die Pfade,

die über Types hin zu lexikalischem Wissen und nutzerdefinierten Konzepten führen. Die Graphentraversierung ermöglicht es, über diese Pfade nicht nur unmittelbare Nachbarschaft als Kontext aufzufassen, sondern auch weiter entfernt liegende und nur indirekt verknüpfte Elemente als Kontext zu betrachten.

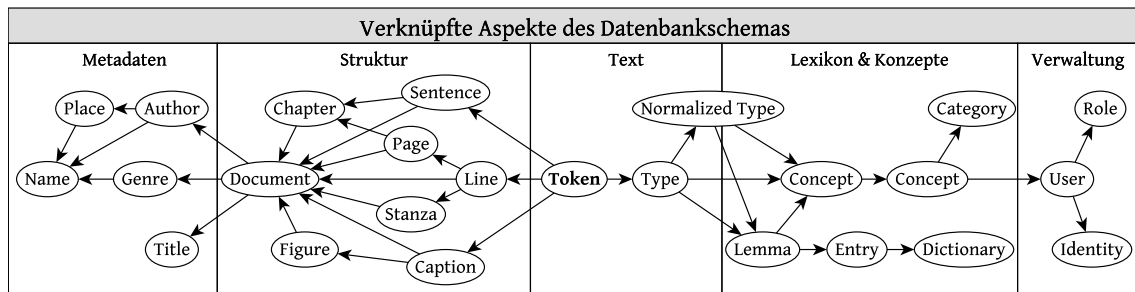


Abbildung 3.2: Schema der tokenzentrierten Datenmodellierung mit möglicher Definition von „Schichten“ innerhalb der verknüpften Umgebung

Im gesamten Modell werden immer wiederkehrende Aspekte wie Sequenz und Hierarchie grundsätzlich immer mit den gleichen Kantentypen abgebildet. Ein Element ist über eine ausgehende next-Kante mit seinem Nachfolger (bzw. seinen Nachfolgern) verknüpft und über eine ausgehende belongs_to-Kante mit seinem übergeordneten Element (bzw. mit mehreren). Für die genauen Ausprägungen dieser kontextdefinierenden Zugehörigkeiten im Modell existieren verschiedene Varianten. Aus diesen ergeben sich jeweils andere Implikationen für die generellen Traversierungsmöglichkeiten und für die Herstellung von effizienter Referenzierbarkeit einzelner Textstellen. In [Abbildung 3.3 auf der nächsten Seite](#) sind die verschiedenen denkbaren Varianten abgebildet. Dazu wird in [Tabelle 3.1 auf Seite 79](#) eine Übersicht über die daraus resultierenden Auswirkungen auf Speicherplatzbedarf und Zugriffsaufwände gegeben.

Zur Auswahl einer „korrekten“ Modellierungsart wird in dieser Arbeit keine Festlegung getroffen. Letzten Endes handelt es sich dabei um eine in der Informatik übliche Abwägung zwischen Speicherplatzbedarf und Bearbeitungsgeschwindigkeit. Dabei sind nicht nur die Komplexitätsklassen, sondern insbesondere auch die konstanten Faktoren zu berücksichtigen und so die Varianten anwendungsfallbasiert gegeneinander abzuwägen. Innerhalb dieser Arbeit wird mit (c), dem vollständigsten Modell, das ohne Positions-Kanten-Properties auskommt, gearbeitet. Eine Abweichung davon macht gewisse Änderungen an Basisabfragen nötig (bzw. Optimierungen möglich), welche sich sämtlichst in Kadmos problemlos vornehmen lassen.

Eine explizite Nummerierung ist insbesondere dann interessant, wenn für viele Opera-

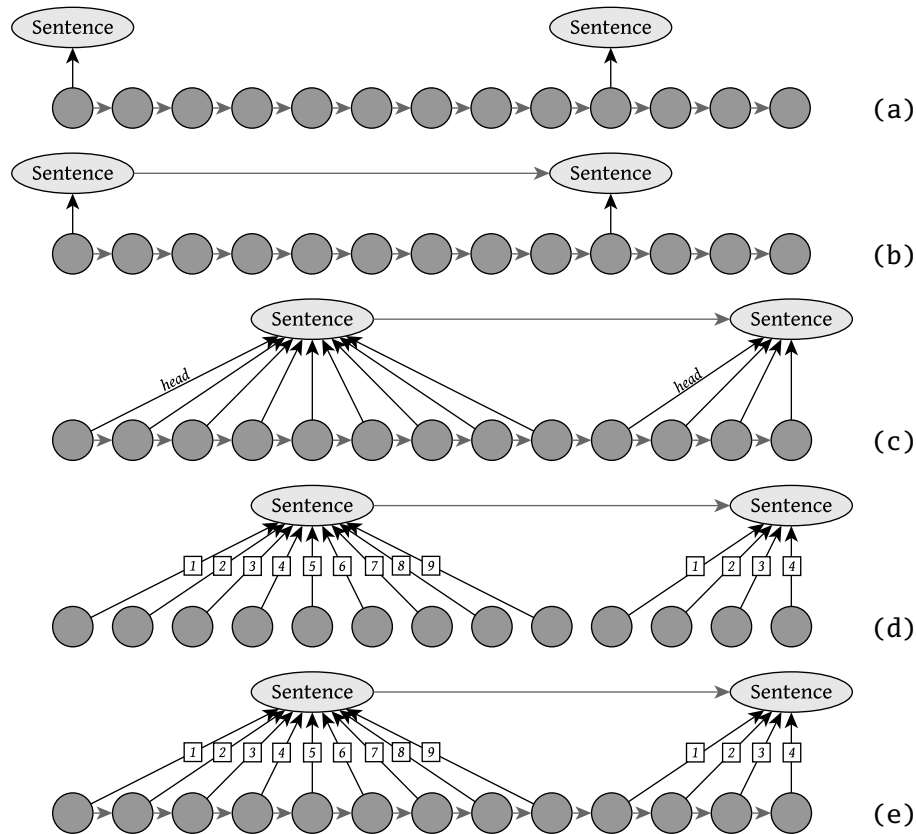


Abbildung 3.3: Verknüpfungsvarianten für Hierarchie und Sequenz: (a) minimal, (b) mit Struktursequenzen, (c) direkte Hierarchisierung, (d) Hierarchisierung ohne Tokenssequenz und (e) vollständig

tionen ein Überblick über die Lage von Fundstellen zueinander gewonnen werden soll, wie es z. B. in [NRR⁺12] als Grundlage für komplexere Zugriffsszenarien verwendet wird („combining the table of contents with semantic tagging, index items, and free-text searching“).

Zusätzlich zu den Varianten zur Modellierung von direkten Hierarchie- und Sequenzbeziehungen können verschiedene Formen der Verknüpfung über Hierarchiestufen hinweg eingeführt werden, um die Traversierungsgeschwindigkeit zu erhöhen. Dies wird beispielhaft in Abbildung 3.4 auf Seite 80 gezeigt. Zur Optimierung von Laufzeiten kann zusätzlich überlegt werden, eigene Kantentypen für die „Abkürzungen“ zu hohen Hierarchiestufen (wie Dokumenten) einzuführen (im Gegensatz zum generischen „belongs_to“), oder den Knotentyp des Zielknotens als Kantenproperty zu speichern. Das tatsächliche Optimierungspotential ist dabei allerdings vom Anwendungsfall und vom verwendeten Backend abhängig. Auch hierzu soll daher keine allgemeingültige Empfehlung abgegeben werden.

	(a)	(b)	(c)	(d)	(e)
 Speicherbedarf					
Knoten	$s+t$	$s+t$	$s+t$	$s+t$	$s+t$
Kanten (Sequenz)	$t-1$	$s+t-2$	$s+t-2$	$s-1$	$s+t-2$
Kanten (Hierarchie)	s	s	t	t	t
Kanten (Gesamt)	$s+t-1$	$2s+t-1$	$s+2t-2$	$s+t-1$	$s+2t-2$
Kantenproperties	–	–	s	t	t
 Zugriffsoperationen	(unter Annahme eines konstanten Index-Lookups)				
zu Vorgänger/Nachfolger-Token	1	1	1	2	1
Tokenfenster der Breite 7 ermitteln	6	6	6	7	6
Token zu Satz	$2i-1$	$2i-1$	1	1	1
Satz zu n-tem Token	n	n	n	1	1
Token zu n-tem Token (selber Satz)	$i+n-2$	$i+n-2$	$i+n-2$	2	2
analog, übernächster Satz	$2k-i+n-1$	$i+n+2$	$n+3$	4	4
letztes Token im selben Satz	$i+n+2$	$i+n-2$	$i+n-2$	2	2
Legende:	s – Satzanzahl t – Tokenanzahl i – aktuelle Token-Position im Satz k – durchschnittliche Länge eines Satzes in Token				

Tabelle 3.1: Speicherbelegung und Abfrageaufwände für verschieden Grade der Verknüpfung im Textdatenmodell

3.3 Technologie und Systemarchitektur

Um das beschriebene Datenmodell in der Rechercheanwendung effizient nutzen zu können, ist die Wahl eines geeigneten Graphdatenbanksystems (als zentraler Komponente für Datenhaltung und -zugriff) von großer Bedeutung. Wie bereits in Abschnitt 2.3.4 auf Seite 56 vorgestellt, existiert mit Apache TinkerPop eine universelle Schnittstelle für Property-Graph-Systeme in einer Java-Umgebung. Die objektorientierte Programmiersprache Java verfügt in Verbindung mit ihrer dynamischen Ausführungsumgebung (Java Virtual Machine) über nützliche Funktionen, wie Just-In-Time-Kompilierung und automatische Speicherverwaltung (Garbage Collection) und ermöglicht zudem eine weitestgehend plattformunabhängige Systementwicklung. Daher wurde Java auch als Ausführungsumgebung für Kadmos bevorzugt.

Kadmos soll auch Korpora unterstützen, deren digitale Repräsentation größer ist, als der zur Verfügung stehende Hauptspeicher. Daneben soll das System schnell gestartet

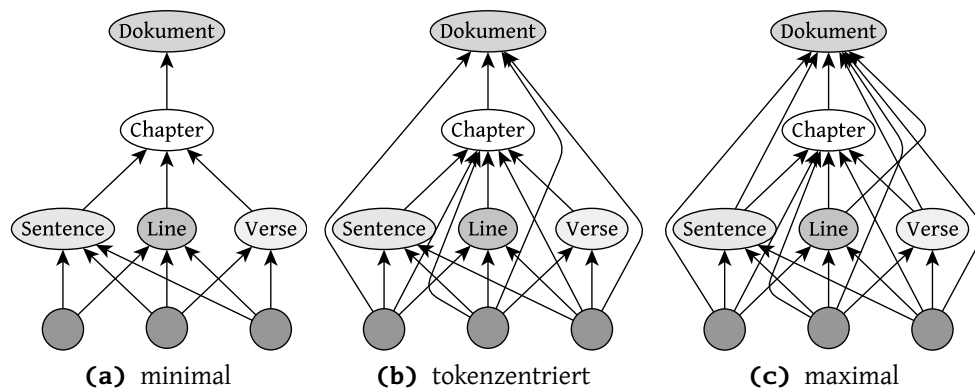


Abbildung 3.4: Verknüpfung von Knoten für direkte Hierarchiesprünge

werden können, ohne lange initialisierungsbedingte Wartezeiten aufzuweisen. Das Datenbanksystem muss daher eine persistente Speicherung der Korpusdaten sowie der nutzerspezifischen Änderungen und Ergänzungen erlauben. Deshalb können reine In-Memory-Lösungen, wie das in [JR13] beschriebene „imGraph“, trotz hoher Abfragegeschwindigkeiten nicht verwendet werden.

Aus der Reihe potentieller Datenbanklösungen wurde das System Titan³ ausgewählt, wobei auf die Vielzahl von Alternativen an dieser Stelle nicht im Detail eingegangen werden kann. Als populärstes System ist hauptsächlich Neo4j⁴ (das möglicherweise reifste Produkt auf dem Markt) hervorzuheben⁵. Im Gegensatz zu diesem weist Titan jedoch den Vorteil auf, unterschiedliche, grundsätzlich austauschbare Speicher-Backends zu unterstützen. Noch in [Sri11] wurden für die Klassifizierung von Graphdatenbanken anhand ihrer Speichersysteme nur zwei Gruppen unterschieden: die Systeme, die auf relationalen Datenbanken aufsetzen und diejenigen mit einem nativem (und fest eingebauten) Graphen-Speicher. Titan hat eine dritte Gruppe begründet, in der NoSQL-Systeme zur Speicherung herangezogen werden.

Diese umfassen z. B. auch massiv verteilbare NoSQL-Datenbanken (mit weniger komplexen Datenmodellen), was für eine hohe Flexibilität hinsichtlich der Skalierung des Systems auf größere verteilte Rechnerarchitekturen sorgt. Verschiedene solcher Systeme bedienen dabei unterschiedliche Bereiche des durch das CAP-Theorem aufgespannten Raums.

³<http://titan.thinkaurelius.com/>

⁴<http://neo4j.com/>

⁵Das Unternehmen hinter Neo4j, die Neo Technologies Inc., richtet viele branchenweite Veranstaltungen, z.B. die internationale Konferenz GraphConnect (<http://graphconnect.com/>) aus. Damit fördert sie die allgemeine Entwicklung dieser Technologie, positioniert das eigene Produkt allerdings auch geschickt als Vorreiter-Lösung.

Neben der Partitionierbarkeit steht bei Apache HBase⁶ die Konsistenz im Vordergrund, während es bei Apache Cassandra⁷ die Verfügbarkeit ist – darüber hinaus ähneln sich beide *Wide Column Stores* allerdings sehr stark.

Als „Zwischenschicht“ zwischen Anwender und Speichersystem unterstützt Titan eine große Untermenge der in der Blueprints-API definierten Funktionen. Es ist also grundsätzlich möglich, auch das Datenbanksystem selbst nachträglich durch ein anderes mit Blueprints kompatibles System auszutauschen. In der Praxis unterscheiden sich alle Systeme allerdings leicht in Details der Konfiguration und Ansteuerung. Insbesondere die Mechanismen zur Index-Definition sind bei Titan sehr systemspezifisch, weshalb ein Austausch nicht ohne zusätzliche Aufwände zu realisieren wäre – ein Umstand, der wahrscheinlich bei allen Systemen dieser Komplexität auf die ein oder andere Weise festzustellen wäre.

Titan besitzt (auch bereits mit einem lokalem Speicher-Backend) eine gute Abfrageperformance sowie die angesprochene gute Skalierbarkeit in Clustern, s. z. B. [JV13]; wobei das System (im Vergleich damaliger Versionen) auch laut [KSM13] nicht das schnellste der untersuchten war. In [MEP+14] wird Titan mit anderen Graphdatenbanken, aber auch mit In-Memory-Paketen ohne Persistenz-Funktionalität verglichen und dabei im Hinblick auf die Antwortzeiten für vorgegebene Abfrageszenarien im Mittelfeld geführt. Es wird dort allerdings als das langsamste bei Daten-Updates beschrieben. Da in Kadmos der lesende Zugriff auf Daten im Vordergrund steht, kann diese Einschränkung jedoch hingenommen werden.

In Anbetracht dieser zur Verfügung stehenden Komponenten ist Java eine adäquate Laufzeitumgebung für Kadmos. Als Programmiersprache für prototypische Entwicklung ist Java allerdings nur bedingt geeignet. Dynamische Programmier- und Scriptsprachen, wie Python oder Ruby, sind grundsätzlich deutlich flexibler einsetzbar. Eine bei deren Verwendung erforderliche separate Installation von Rechercheanwendung und Datenbanksystem führt allerdings zu nicht unerheblichen prozessübergreifenden Kommunikationsaufwänden zwischen den Komponenten. Die im Rahmen dieser Arbeit vorgestellte Lösung verwendet JRuby⁸, eine in Java geschriebene Implementierung der Programmiersprache Ruby. Sie kann gemeinsam mit den auf TinkerPop basierenden Datenbankkomponenten in der selben Laufzeitumgebung ausgeführt werden. Die Datenverarbeitung kommt so

⁶<http://hbase.apache.org/>

⁷<http://cassandra.apache.org/>

⁸<http://jruby.org/>

ohne zusätzliche Serialisierungs- und Kommunikationsaufwände aus.

Für Ruby stehen zahlreiche gekapselte Programmbibliotheken zur Verfügung (sogenannte *Ruby Gems*), von denen die meisten ohne Anpassungsaufwände auch unter JRuby genutzt werden können. Gems werden u. a. in einem zentralen Repository in aktuellen und vergangenen Versionen vorgehalten. Über das Gem „Bundler“⁹ können alle Gem-Abhängigkeiten eines Projektes deklarativ erfasst und automatisch in den neuesten der konfliktfrei kombinierbaren Versionen heruntergeladen und für das Projekt installiert werden.

Abbildung 3.5 zeigt den prinzipiellen Aufbau des Systems. Die gestrichelten Linien sollen andeuten, dass als Speicher-backends und Volltextindex-Systeme sowohl externe, auf Rechnercluster skalierbare Lösungen als auch eingebettete Komponenten verwendet werden können. Diese werden direkt von Titan entsprechend der Konfigurationsdatei von Kadmos angesteuert. Durch Kadmos wird auch der angesprochene TinkerPop-Stack geladen, dessen generische Abfragemechanismen dann für die JRuby-Komponenten zur Verfügung stehen. Als eingebetteter Webserver wird der Java-Applikationsserver „Glassfish“¹⁰ über das JRuby-Gem „Mizuno“¹¹ geladen und mit der *Middleware*-fähigen universellen Webserver-Schnittstelle „Rack“¹² verknüpft. Über dieses Konstrukt kann die *Threading*-Funktionalität von Glassfish in JRuby genutzt werden sowie eine synchrone Anfragebeantwortung erfolgen.

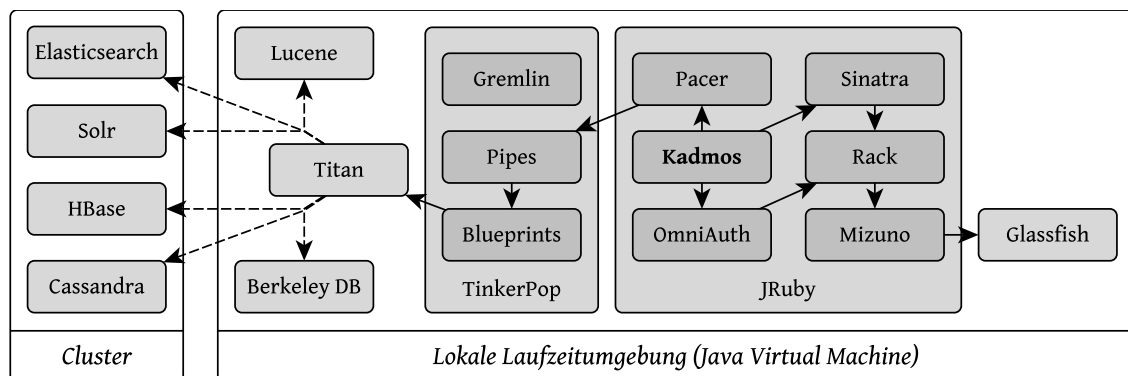


Abbildung 3.5: Architekturentwurf mit lokalen und entfernten Komponenten der Kadmos-Umgebung

Die Rack-Middleware wird von mehreren weiteren Gems verwendet. Über den webba-

⁹<http://bundler.io/>

¹⁰<http://glassfish.java.net/>

¹¹<http://github.com/matadon/mizuno>

¹²<http://rack.github.io/>

sierten Login-Mechanismus von „OmniAuth“¹³ können viele verschiedene Authentifizierungssysteme verwendet werden, um Accountinformationen fremder Identitätsprovider in Kadmos zu nutzen. Diese werden internen Nutzerkonten zugeordnet, die im Graphdatenbanksystem hinterlegt werden. Das Webframework „Sinatra“¹⁴ setzt ebenfalls auf der Rack-Funktionalität auf. Sinatra steuert das Anfrage-Routing, verwaltet die von außen erreichbaren APIs und beantwortet Anfragen von Webbrowsern mit dynamisch erzeugten HTML-Seiten. Kadmos enthält noch viele weitere interne Komponenten und eingebundene Gems, auf die an dieser Stelle nicht einzeln eingegangen werden kann. Von zentraler Bedeutung ist jedoch die Komponente, die die effiziente Interaktion mit der Graphdatenbank ermöglicht:

Wie bereits in Abschnitt 2.3.6 auf Seite 62 vorgestellt, existieren viele verschiedene Ansätze für die Abfrage graphförmiger Daten. Dadurch, dass die Textdatendomäne, so wie hier modelliert, einen klar umrissenen Aufbau mit gut abschätzbaren Eigenschaften besitzt, ist die Wahl einer imperativen Abfragesprache möglich. Für Ruby steht mit dem Gem „Pacer“¹⁵ eine stark an Gremlin orientierte Abfragesprache zur Verfügung, welche ihre Sprachkonstrukte direkt in JRuby integriert. Dabei werden als sogenannte *Routes* Datenverarbeitungsketten (über TinkerPop Pipes) erstellt, welche dann effizient ausgeführt werden können. Zur Erzeugung der Routen existiert eine *Domain Specific Language (DSL)* die über *Methoden-Chaining* deren schrittweisen Aufbau durch Traversierungsschritte, Filterung, *Lookaheads*, Schleifen, Bedingungen, etc. unterstützt. Die Routen werden dann, wenn auf ihre Ergebnisse zugegriffen wird, „lazy“ im Blueprints-Framework evaluiert.

Die Eigenschaften der gewählten Laufzeitumgebung und Systemarchitektur kommen der prototypzentrierten Entwicklung dabei sehr entgegen: Java unterstützt *Just-in-time*-Kompilierung, kann also neuen Quelltext direkt zur Laufzeit in effizient ausführbaren Maschinencode übersetzen, wodurch die Möglichkeit zum Hinzufügen von Programmfunktionalität ohne Geschwindigkeitseinbußen (im Vergleich zum interpretierten Fall reiner Scriptsprachen) besteht. JRuby erlaubt das Wiederöffnen von Klassen, so dass auch bestehende Funktionalität zur Laufzeit angepasst werden kann. Daneben unterstützt es eine Introspektion von beliebigen Instanzen. In Kombination mit den von Pacer bereitgestellten Funktionen bedeutet das, dass zur Laufzeit eine Analyse von Abfrage-Routen und deren konkreten Ausführungsstrategien stattfinden kann, was zur Geschwindigkeitsoptimierung unerlässlich ist.

¹³<http://github.com/omniauth/omniauth>

¹⁴<http://www.sinatrarb.com/>

¹⁵<http://github.com/pangloss/pacer>

Über eine einfache aus Ruby-Konstrukten aufgebaute Definitionssprache können Knoten-, Kanten- und Property-Typen beispielsweise so definiert werden:

```
1 node_property :string_value, :string, fulltext:true
2 node_type :name
3 edge_type :name
4 edge_property :name_type, :integer
5 edge_type :next
6 edge_type :belongs_to
7 edge_property :sequence_number, :long
```

Quelltext 3.1: Beispielcode für die Schemadefinition in Kadmos

Das Schema von Graphdatenbanken kennt, wie erwähnt, keine Knotentypen und erzwingt daher auch keine festen Typen als Start- oder Endknoten einer Kante bestimmten Typs. Ebenso wenig wird eine Zuweisung von Kantenproperties zu Kantentypen vorgenommen. Diese Modellierungsfreiheit wird für die Schemadefinition in Kadmos übernommen. Die obenstehenden Methodenaufrufe erzeugen lediglich neue numerische Knotentyp-IDs (als effizient zu speichernder Wert für die `node_type`-Property), Kantentypen und Indexstrukturen für Properties. Daher muss für letztere zusätzlich ein Datentyp angegeben werden. Weitere Wertebereichseinschränkungen, wie z. B. `uniq` für die Erzwingung eindeutiger Werte, sind möglich. Die Schemadefinition erzeugt außerdem automatisch domänenspezifische Abfragerouten für die Nutzung in Pacer-Abfragen, wie `create_kantentyp_node` oder `find_kantentyp_node`.

Kadmos ist als ein Recherchesystem mit Fokus auf Backendfunktionalität nicht ausschließlich für die direkte Nutzung über eingebaute Bedienoberflächen konzipiert (auch wenn diese in begrenzter Zahl bereits integriert sind), sondern ist insbesondere auf das Bereitstellen von über Netzwerkverbindungen nutzbaren Diensten ausgelegt. Die dafür getroffenen Vorkehrungen und eingebauten Kommunikationsmechanismen werden im nächsten Abschnitt vorgestellt.

3.4 Asynchrone Webservicearchitektur

Webservices sollen den Zugriff auf die Funktionalität und Ressourcen eines Systems über etablierte Transportmechanismen und Protokolle des [WWW](#) ermöglichen. Sie stellen eine grundsätzlich system- und implementierungsunabhängige Schnittstelle dar, die sich für eine lose gekoppelte Interaktion zwischen Programmen und Programmteilen eignet.

Traditionell stellen Webservices eine komplexe, mehrschichtige Webtechnologie dar, die etwa das [Simple Object Access Protocol \(SOAP\)](#) für [XML](#)-basiertes *Messaging* als Basis für Schnittstellenbeschreibungen nach der [Web Service Description Language](#) bzw. [Web Service Definition Language \(WSDL\)](#) nutzt. Darüber hinaus existieren Protokolle zum automatisierten und maschinenlesbaren Veröffentlichen und Auffinden von Services in Netzwerken, begleitet von einer darüber liegenden Schicht, die den „*Service Flow*“, also die Kombination und Orchestrierung mehrerer Teilservices betrifft und Protokolle, wie ([Web Services](#)) [Business Process Execution Language \(BPEL\)](#) enthält, vgl. [\[Sha08\]](#).

In Kadmos sollen möglichst einfache Varianten von Webservices zum Einsatz kommen, die schnell und unkompliziert von *Clients* genutzt werden können. Endpunkte sollen dabei grundsätzlich ohne die Notwendigkeit zusätzlicher Abstraktionsschichten direkt verwendet werden können – etwa über einen einfachen Adressaufruf über Browser oder über Standardsoftware auf der Betriebssystem-Kommandozeile.

Die Kommunikation zwischen zwei Endpunkten im Internet wird über verschiedene übereinanderliegende Abstraktions- und Protokollschichten geregelt, s. z. B. [\[Sha08\]](#). Die genaue Einteilung von Protokollen in Schichten ist vom Referenzmodell abhängig, wobei häufig das sieben Ebenen unterscheidende Modell der [Open Systems Interconnection \(OSI\)](#) verwendet wird. Zwischen den hardwarenahen Schichten und den Protokollen der Anwendungsebene findet mit dem [Internet Protocol \(IP\)](#) ein paketvermittelnder, teilnetzübergreifend adressierender Versand von Teildaten über dynamische Routen statt. Um eine vollständige und fehlerfreie Übermittlung und die korrekte Reihenfolge der Pakete zu gewährleisten, überwacht und reguliert das [Transmission Control Protocol \(TCP\)](#) den Datenfluss durch zusätzliche Steuerpakete. Dabei werden durch regelmäßige Kontaktmeldungen zwischen den Kommunikationspartnern virtuelle [TCP](#)-„Verbindungen“ aufrechterhalten, innerhalb derer eine effiziente Kommunikation möglich ist. Der Aufbau einer solchen Verbindung benötigt eine gewisse Zeitspanne.

Das [Hypertext Transfer Protocol \(HTTP\)](#) ist ein statusloses Protokoll, das üblicherweise [TCP](#) für den Datenaustausch verwendet. Es erlaubt einem *Client* (der meist ein Webbrowser ist), Anfragen (*Requests*) an einen Rechner zu senden, welcher einen [HTTP](#)-Dienst anbietet und mit *Responses* antwortet. Dieses Protokoll wird für leichtgewichtige Schnittstellen nach den Prinzipien des „[Representational State Transfer \(REST\)](#)“, vgl. [\[Fie00\]](#) verwendet. Auch diese Schnittstellen werden in statusloser Kommunikation genutzt. Das hat beispielsweise den Vorteil, dass ein *Load Balancing* mittels Verteilens der Anfragenlast auf mehrere identische Systemkopien damit einfach umzusetzen ist, ohne, dass eine

bestehende „Kommunikationshistorie“ unter den Servern ausgetauscht werden muss.

In Kadmos sollen über Webservices angesteuerte APIs in der Lage sein, Zwischenstände aufwändiger Berechnungen an den Browser des Nutzers zurückzumelden. Die dafür denkbaren Kommunikationsmechanismen sind in Abbildung 3.6 schematisch abgebildet. Bei [Asynchronous JavaScript and XML \(AJAX\)](#) müssten dabei periodisch Statusabfragen gesendet werden, während Web Sockets eine vom Server ausgehende Meldung unterstützen, die neben der eigentlichen HTTP-Kommunikation abläuft. Schließlich ermöglichen *Server-Sent Events* (mit dem Beinamen „Event-Source“), s. [Hic15], ein gestaffeltes, zeitversetztes Rücksenden des Nachrichteninhalts (im Sinne des *HTTP Response Body*).

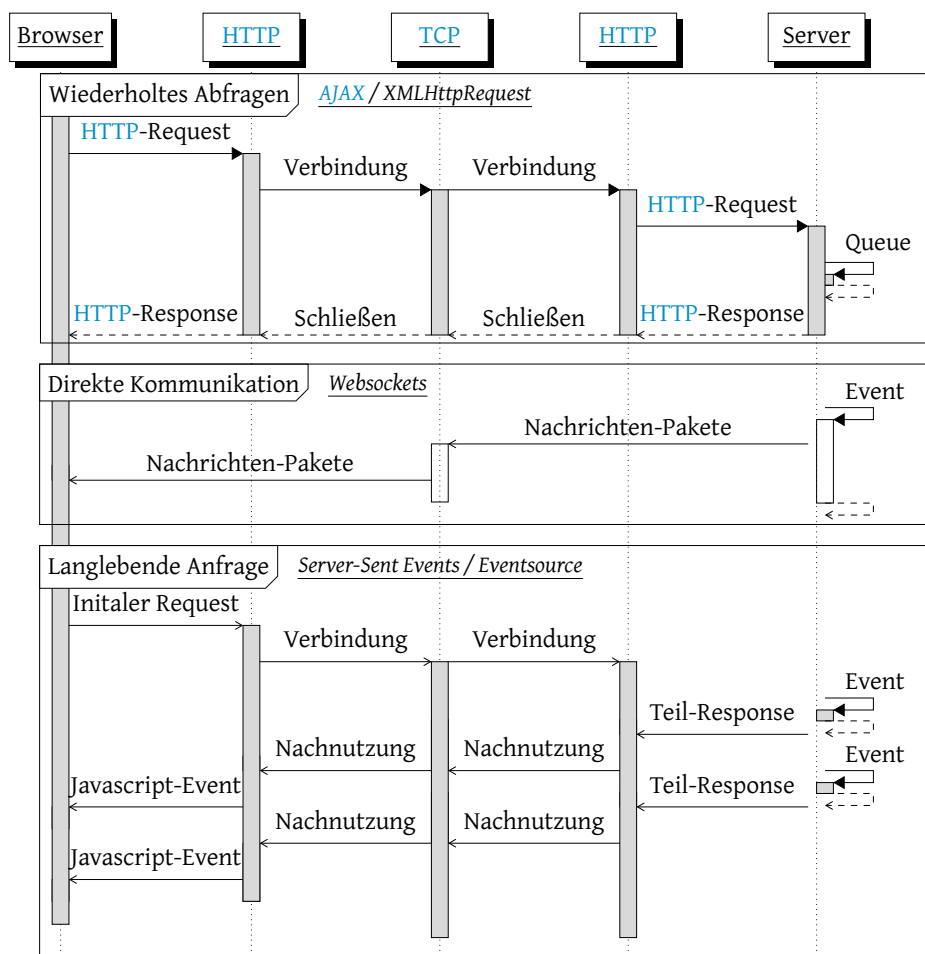


Abbildung 3.6: Verschiedene Modelle der Client-Server-Kommunikation zum asynchronen Nachladen von Inhalten

In Kadmos werden Server-Sent Events genutzt, da sie browserseitig einfach zu handhaben sind und verzögerungsfreie Rückmeldungen ermöglichen. Ein Vorteil gegenüber Websockets ist, dass keine explizite Behandlung von Aufruf-Abbrüchen stattfinden muss, z. B.

wenn das Browserfenster geschlossen wird. Da die „Ereignis-Quelle“ in diesem Anwendungsszenario nicht kontinuierlich sendet, sondern nur anfragespezifische Rückgaben an einzelne aufrufende Clients schicken soll, sollten beim Aufruf im Browser stets [URL-Query-Parameter](#) angefügt werden, um ein automatisches *Pre-Fetching* zu verhindern. Als Ersatzfunktion (*Fallback*) bei Nicht-Unterstützung von Event-Sources im Browser kann ein sogenanntes *Polyfill* genutzt werden. Dieses realisiert die Anfrage als [AJAX](#)-Aufruf, und feuert alle Events beim Schließen der Verbindung. Deshalb lohnt es sich, clientseitig eingehende Zwischenergebnisse für einige Millisekunden zwischenzuspeichern (und auf neuere Zwischenergebnisse zu warten), bevor visuelle Änderungen in der Seitendarstellung angestoßen werden.

In Kadmos sind sowohl klassische synchrone [REST](#)-Abfragen¹⁶ als auch Services mit asynchronen Antworten realisierbar. Diese werden über die selben gekapselten Funktionseinheiten implementiert, so dass ohne Mehrfachaufwände in der Programmierung beide Schnittstellentypen gleichzeitig abgedeckt werden können. Intermediäre (asynchrone) Ausgabe-Aufrufe werden dabei im synchronen Fall nicht versendet, sondern verworfen, so dass die finale Ausgabe als einzige [REST](#)-Antwort zurückgegeben wird. Die Schritte zur Implementierung einer einfachen Schnittstelle werden in Abschnitt [3.10 auf Seite 122](#) vorgestellt. In Abschnitt [C auf Seite 252](#) im Anhang werden Beispielaufrufe bereits vorhandener [API](#)-Endpunkte kurz vorgestellt.

3.5 Datenimport

Der Datenimport in Kadmos erfolgt direkt über Ruby-Programmcode, der innerhalb der laufenden Kadmos-Instanz ausgeführt wird. Dadurch ist ein (je nach gewähltem Backend mehr oder weniger unmittelbar) direktes Arbeiten mit dem partiell importierten Korpus möglich. Es werden verschiedene Basisimporter (etwa für Verzeichnisse voller *Plain-text*-Dateien) und spezialisierte Beispielimporter (etwa für das [REUTERS-Korpus](#) oder [VOYNICH-Korpus](#)) standardmäßig zur Verfügung gestellt. Der Import beliebiger weiterer Formate und Dokumentsammlungen kann über eigens einzurichtende Module erfolgen, welche auf eine umfangreiche Basisfunktionalität der *Importer*-Klasse zurückgreifen können. In den vorliegenden Importmodulen kommen zum Teil bereits recht fortgeschrittene Techniken zum Einsatz. So existiert die Möglichkeit, komprimierte Archivdateien

¹⁶mit der Abwandlung, dass das Antwortformat auf [JSON](#) festgelegt ist – so entfällt die Notwendigkeit einer *Content Negotiation*

aus dem Internet zu laden und diese ohne Zwischenspeicherung auf der Festplatte im Hauptspeicher zu entpacken und die einzelnen enthaltenen Dokumente schrittweise in die Graphdatenbank einzulesen.

Der Einlesevorgang wird durch einzelne Methodenaufrufe gesteuert, die einer „event-basierten“ Abarbeitung entsprechen: Das System besitzt einen internen Zustand, der durch einfache Kommandos wie den Methodenaufruf `line` verändert werden kann – in diesem Fall um das Anlegen eines neuen Zeilen-Strukturelements, zu welchem die dann folgenden Token zugewiesen werden. Der Import kann so „auf Zuruf“ erfolgen und in dieser Konfiguration tatsächlich auch auf der Basis von *Events* oder *Streams* erfolgen. Typischerweise werden die Methodenaufrufe aber in Schleifen ausgeführt, in denen auch die Originalformate ausgelesen werden.

Das in Abbildung 3.1 auf Seite 76 gezeigte Beispieldokument kann programmatisch durch wenige Methodenaufrufe in die Datenbank eingespeist werden, wie im folgenden Quelltext 3.2 gezeigt wird:

```
1 document(string_value: "beispiel.txt").ensure_name("Beispieldokument")
2 author.ensure_name("Unbekannt")
3 chapter
4 line
5 sentence
6 "The quick brown fox jumps over the lazy dog.".split(" ").each{|s| token s}
7 sentence
8 token "All"
9 line
10 "the leaves are brown.".split(" ").each{|s| token s}
11 sentence
12 "The ship leaves tonight.".split(" ").each{|s| token s}
```

Quelltext 3.2: Import eines Dokuments über direkte Methodenaufrufe

Die Tokenisierung kann in Kadmos dabei, wie erwähnt, sehr simpel durchgeführt werden, etwa durch ein Splitten an Leerstellen. Bei Sprach- und Schriftsystemen, in denen Wörter überhaupt nicht oder nicht konsequent mit Leerzeichen getrennt werden (z. B. beim Thailändischen) muss eine entsprechende Segmentierung entweder über die Auswertung existierender Token-Annotationen im zu importierenden Dokument oder aber durch die Nutzung von externen Tokenisierungswerkzeugen während des Importvorgangs hergestellt werden.

Zugehörige Types werden beim Anlegen neuer Token selektiert oder automatisch neu erzeugt und anschließend verknüpft. Im Importer kann ein beliebiger Normalizer angegeben werden, durch welchen die Erzeugung normalisierter Types durchgeführt wird. Diese Funktionalität ist in einer eigenen Klasse gekapselt, damit sie später bei der Abfrage auch zur Normalisierung von Nutzereingaben verwendet werden kann. Auf das Thema der Normalisierung wird in Abschnitt 3.6 auf Seite 91 noch näher eingegangen.

Häufig verwendete Struktureinheiten, wie Zeile und Satz, sind in Kadmos standardmäßig enthalten. Das Datenmodell kann jederzeit um weitere übliche Strukturierungselemente erweitert werden. Die einzelnen logischen Struktureinheiten lassen sich zum Teil jedoch nur schwer aus den zu importierenden Originaldokumenten extrahieren. Dieser Umstand soll am Beispiel des BTL-Korpus, welches in Form von XML-Dateien¹⁷ vorliegt (und so grundsätzlich eine maschinenlesbare Strukturierung aufweist), demonstriert werden.

Da die XML-Repräsentation sich am Ausgabemedium der gedruckten Buchfassung orientiert – es sich dabei also um ein Arbeitsinstrument der so genannten „Druckvorstufe“ handelt – wird die logische Struktur, die vorrangig in Kadmos importiert werden soll, von ausgabebedingtem Markup überdeckt. Für kurze Abschnitte (in der Regel Sätze) erfolgt jeweils eine (im XML in sich geschachtelte, jedoch nicht immer logisch hierarchische) Auszeichnung. Diese wird teilweise durch nummerierte Referenzierungsinformationen ergänzt. Die erste und zweite Beschreibungsebene bilden dabei immer der Autorenname und der Werkstitel. Danach folgen verschiedenste strukturelle Untereinheiten. Zum Teil sind in den Ebenen direkt Kapitelüberschriften oder Ähnliches vermerkt, meist aber nur allgemeine Beschreibungen der Struktureinheit.

In diesen Vermerken von „level_3“, in einigen Fällen bis hin zu „level_8“ herrscht wenig Kohärenz in der Hierarchisierung. Nur zum Teil bezieht sich die jeweilige Untergliederungseinheit auf die der übergeordneten Ebene, ist also tatsächlich in diese zu schachteln, wie z. B. die Zeile eines Paragraphen. Die Angabe von Vers, Brief, Strophe, Paragraph, Zeile, Seite, usw. findet je nach Dokument in unterschiedlichen Ebenen statt und ändert sich an einigen Stellen auch innerhalb eines Dokuments. In Tabelle 3.2 auf der nächsten Seite ist die große Vielfalt der möglichen Angaben ablesbar.

Neben der so geschaffenen heterogenen Schachtelungssituation ist auch festzustellen, dass die fortlaufende Nummerierung nicht immer direkt inkrementell abläuft. Zum einen hat das historische Gründe der Editionspraxis (z.B. spätere Einfügungen in ältere

¹⁷mit dem Dokumenttyp „NCBI Book 3.0“, <http://dtd.nlm.nih.gov/book/3.0/book3.dtd>

Ebene	Häufige Bestandteile der Bezeichner
level_3:	fragm. vers. epigr. elegia carm. p. cap. § lib. decl. incertarum incertorum nota fragmenta paradoxon oratio actio scribae epist. fab. distichon sat. epil. prol. littera Epist. suas. hexastichon incertae vol. ecloga strophalaudes explicit
level_4:	vers. lin. p. cap. fragm. argum. § epist. lib. carm. sat. epigr. fab. contr. sent. sermo
level_5:	lin. p. vers. acrostichis § vol. cap.
level_6:	lin. p. § (p.
level_7:	lin. p.
level_8:	lin.

Tabelle 3.2: Vorkommende Nummerierungspräfixe und allgemeine Abschnittsbezeichner in verschiedenen Hierarchieebenen im [BTL-Korpus](#)

re Nummerierungsschemata) oder eine zu grobe Auflösung (z. B. wenn mehrere Sätze oder Zeilen in der kleinsten untergliederten Einheit untergebracht sind). In solchen für die Nachmodellierung im Graphen schwierigen Fällen bietet sich die Einführung eines neutralen Strukturelements an, das die kompletten Informationen der aktuellen „Hierarchieangaben“ als konkatenierte Zeichenkette speichert und welches als einziges Strukturelement zwischen Dokument und Token positioniert wird. Damit ist eine Referenzierung von Textstellen wie im gedruckten Buch möglich und der Analyseaufwand des zu importierenden Dokumentenmodells wird verringert. Diese Lösung ist nicht ideal, da aussagekräftige Statistiken zu Satzlängen oder Ähnlichem dann nicht mehr möglich sind, aber als intermediäre Lösung im Sinne des *Prototypings* ist dieser Ansatz hervorragend geeignet.

Neben der Übernahme von Textdaten mit ihren verschiedenen strukturellen Konfigurationen wird bei Importvorgängen in Korpusverwaltungssystemen üblicherweise auch die Zusammenführung und Korrektur von Metadaten vorgenommen. Das flexible Datenmodell und der Verzicht auf Vorberechnungen ermöglichen es jedoch, diesen Aspekt in Kadmos vollkommen losgelöst vom Import zu betrachten. Im laufenden Recherchebetrieb und als Teil der Korpusexploration können die Metadaten jederzeit angepasst werden, wobei manuelle Arbeiten durch etablierte Datenintegrationsmodelle¹⁸ ergänzt werden können, sobald externe Identifier, z. B. von Normdatenprovidern oder aus dem

¹⁸Hierbei ist besonders das repräsentantenbasierte Modell von Topic Maps, das u. a. in *Subject Identifiers* und *Subject Locators* unterscheidet, hervorzuheben.

LOD-Umfeld verwendet werden.

Beim Import von Textdaten sollten diese im System in einer Form repräsentiert werden, die einen intuitiven Zugriff auf Zeichenkettenebene ermöglicht. Um angesichts der angesprochenen Kodierungsvielfalt für Zeichensätze einen solchen Zugang zu ermöglichen, findet oft bereits beim Import eine Normalisierung der Texte statt.

3.6 Zeichennormalisierung

Normalisierung ist – generell gesprochen – ein Verfahren, mit dem Varianten in der Form einer Äußerung verringert werden sollen, um die Kernäußerung besser isoliert betrachten, verarbeiten und quantitativ auswerten zu können. Es findet dabei eine Umformung aller Elemente der Eingabe statt, die mengentheoretisch betrachtet als Abbildung angesehen werden kann¹⁹. Bei der Normalisierung soll erreicht werden, dass im Wesentlichen Gleichbedeutendes die gleiche Repräsentation erfährt und Unterschiedliches auch in der Abbildung unterscheidbar bleibt.

Praktischen Nutzen hat die Normalisierung bei nicht-bitgenauen Vergleichen von Zeichenketten. Soll für eine Textsuche beispielsweise die Groß- und Kleinschreibung eines Begriffs ignoriert werden, bietet es sich an, bei der Indizierung des Textes eine komplette Umwandlung in Klein- oder Großbuchstaben vorzunehmen. Die selbe Operation muss anschließend auch auf die Anfrage angewendet werden. Der Satz „Heute ist ein schöner Tag!“ könnte zu „HEUTE IST EIN SCHÖNER TAG!“ normalisiert werden. Ein gesuchtes Wort „tag“ würde die Zeichenfolge „TAG“ ergeben, welche sich dann bitgenau im indizierten Text wiederfinden lässt. Eine weitere Anforderung könnte die transparente Behandlung von diakritischen Zeichen sein, wonach „Hallo“, „Hàllo“ und „Hálla“ einen gemeinsamen Normalisierungsrepräsentanten erhalten sollen, welcher sich allerdings von dem anderer ähnlicher Wörter, wie „Hello“ unterscheiden muss.

Eine einzige Normalisierungsregel, die alle Vorkommen von „a“ und „à“ durch „á“ ersetzt und alle anderen Zeichen gleich belässt, würde für dieses Beispiel bereits eine valide Normalisierung darstellen. Denkbar wäre auch, „a“, „à“ und „á“ durch „x“ zu ersetzen, oder ganz zu löschen. Die „Halla“-Varianten würden dabei zu „Hxlla“ oder „Hllo“, und „Hello“ bliebe erhalten. Es ist jedoch leicht einzusehen, dass bei solchen, die Funktion der

¹⁹Da die Bildmenge in der Praxis meist keine „ungenutzten“ Elemente enthält, ist die Normalisierung ebenso eine surjektive Relation.

Zeichen für die Sprache nicht berücksichtigenden Normalisierungsansätzen bei einer Erweiterung des Vokabulars sehr schnell Probleme auftreten können. Die „x“-Substitution würde z. B. „Boa“ und „Box“ auf die selbe normalisierte Form abbilden, was ganz offensichtlich nicht gewünscht ist. Ein guter Substituent lässt sich oft aus der Menge der graphisch verwandten bzw. lautlich oder im Verwendungskontext „ähnlichen“ Zeichen finden.

Dadurch wird auch eine manuelle Inspektion der Normalisierung vereinfacht, die im Vergleich zur Abbildung auf Nominalzahlen oder abstrakte Zeichen²⁰ eine nach wie vor „lesbare“ Variante des Ursprungstextes abbildet.

Die Bandbreite von potentiell zu normalisierenden Phänomenen in der Verschriftlichung ist groß. Im Folgenden werden als Beispiel aus der Praxis einige der Besonderheiten vorgestellt, die die Normalisierung historischer Dokumente in nicht-lateinischen Alphabeten mit sich bringen kann. Als Grundlage werden Wortkodierungen aus den Editionen altgriechischer Texte des [PAPYRI-Korpus](#), verwendet.

Abbildung 3.7: Auftretende Unicode-Zeichen in digitalen Editionen altgriechischer Texte, sortiert nach Codepoint-Nummer

Abbildung 3.7 zeigt die Fülle der in den Texten vorkommenden Zeichen. Es werden über 1600 Unicode-Endpunkte²¹ zur digitalen Kodierung der Texte verwendet. Während einige der außergewöhnlichen Zeichen, etwa die Tierkreiszeichen Υ bis H , durchaus im Rahmen ihrer in Unicode erfassten Bedeutung verwendet werden, wurden andere offenbar nur wegen ihrer äußeren Erscheinungsform eingefügt, wie der mathematische Operator $\otimes$ (U+2297). Als doppelte, öffnende, spitze Klammer wurde stellenweise ebenfalls die der mathematischen Notation vorbehaltene Variante (U+27EA) statt der neutralen (U+300A)

²⁰Diese könnten etwa in den privaten Unicode-Bereichen (*Private Use Areas*) anwendungsspezifisch definiert werden.

²¹inklusive verschiedener Leerzeichen (*Whitespaces*) und sonstiger undruckbarer Zeichen

verwendet. In Abbildung B/1 auf Seite 246 im Anhang sind die Zeichen noch einmal nach der Häufigkeit ihres Auftretens sortiert angegeben.

Für zahlreiche griechische Minuskeln²² existieren Alternativ-Glyphen, die im Grunde den selben Buchstaben beschreiben, durch die aber eine Formvariante codiert werden soll: für Kappa $\kappa \mapsto \text{ϰ}$ (U+03F0), Epsilon $\epsilon \mapsto \text{ϵ}$ (U+03F5), Theta $\theta \mapsto \text{ϑ}$ (U+03D1), Rho $\rho \mapsto \text{ϱ}$ (U+03F1) sowie Phi $\phi \mapsto \text{ϕ}$ (U+03D5). Im kompletten Unicode-Standard existieren darüber hinaus derzeit²³ über 20 Codepunkte, die ein griechisches oder koptisches Phi als Text- oder Formelzeichen repräsentieren²⁴. Beim End-Sigma ς ist die Formvariante nicht nur stilistischer Natur, sondern fest an die letzte Position im Wort gebunden.

Auch diakritische Zeichen sind oft in mehreren Varianten vorhanden: Es existieren typographisch bewusst identisch angelegte Zeichen für einen prinzipiell gleichen Verwendungskontext, die sich doch in ihrer Bedeutung unterscheiden können: Eine Verwendung des *Oxia* (z.B. U+1FFD) zeichnet eine Änderung in der Tonhöhe im polytonischen Griechisch (im wesentlichen Altgriechisch) aus, wohingegen die Nutzung des *Tonos* (z.B. U+0384) für die Betonung im monotonischen Griechisch steht. Beide sollten identisch dargestellt werden, es existieren jedoch sowohl unterschiedliche Einzelsymbole und kombinierende Symbole, als auch zwei komplett eigenständige Sammlungen damit vorkombinierter Symbole für das gesamte griechische Alphabet in Unicode.

Für den Hauchlaut-Indikator für Vokale und Diphthonge (sowie den Buchstaben Rho) am Wortanfang (*Spiritus asper*) existieren drei korrekte Kodierungsvarianten²⁵. Hierbei sind bei der Darstellung der kodierten Texte bestimmte Regeln zur Positionierung der Indikatoren (bei Kombination mit anderen diakritischen Zeichen sowie die Verschiebung vor Großbuchstaben oder auf Folgevokale) zu beachten. Teilweise wird versucht, dieses Verhalten durch Ändern der Zeichenreihenfolge im Vorfeld der Darstellung zu erzwingen, was nicht immer standardkonform ist.

Aus der Vielfalt aller möglichen Permutationen dieser Codepunkte ergibt sich eine enorme Varianz unterschiedlicher valider Kodierungsweisen. Das Griechische ist dabei unter den historischen Sprachen kein besonderer Extremfall. Viele komplexe Regeln, Varianten

²²Damit sind Kleinbuchstaben gemeint, wobei in vielen Fällen nicht in den Manuskripten sondern erst in den Editionen zwischen Groß- und Kleinschreibung unterschieden wird.

²³bezogen auf Version 9 vom 21. Juni 2016

²⁴U+0278, U+03A6, U+03C6, U+03D5, U+1D60, U+1D69, U+1DB2, U+2C77, U+1D6BD, U+1D6D7, U+1D6DF, U+1D6F7, U+1D711, U+1D719, U+1D731, U+1D74B, U+1D753, U+1D76B, U+1D785, U+1D78D, U+1D7A5, U+1D7BF und U+1D7C7

²⁵U+0314 (*Combining Reversed Comma Above*), 02BD (*Modifier Letter Reversed Comma*) und U+1ffe (*Greek Dasia*)

und Abhängigkeiten existieren auch in anderen Alphabeten – durchaus auch in denen moderner Sprachen.

Dieser kleine Exkurs in die praktischen Probleme der Zeichennormalisierung bringt die Erkenntnis, dass für diese Aufgabe keine generische Lösung entwickelt werden kann. Die Datenaufbereitung in Kadmos ist daher als ein frei anpassbares Verfahren angelegt, welches für den Import verschiedene vordefinierte Normalisierungs-Komponenten bereithält, daneben aber auch die Implementierung eigener „Normalisierer“ ermöglicht. Für die einfache Behandlung von diakritischen Zeichen im Standardnormalisierer wird zunächst die NFD gebildet. Anschließend werden alle Zeichen entfernt, die selbst keine horizontale Ausdehnung besitzen (*nonspacing marks*, wie kombinierende und modifizierende Zeichen). Es ist leicht einzusehen, dass dabei fürs Deutsche eine spezielle Behandlung von Umlauten erfolgen sollte, oder bei der Verwendung dieser Normalisierungsstrategie zuvor eine Umwandlung von Umlauten in Diphthonge angeraten ist. In Kadmos können solche sprachabhängigen Funktionen in entsprechenden abgekapselten Teilmodulen vorgehalten werden.

Es können grundsätzlich auch mehrere Normalisierungsvarianten parallel im Modell erfasst werden, was natürlich bei Wortfrequenz-basierten Statistiken nach der Traversierung des Modells berücksichtigt werden muss. In diesem Fall empfiehlt es sich, an die Kante zwischen Type und normalisiertem Type die Normalisierungsart in Form einer speziellen, mit entsprechendem Wert versehenen Kanten-Property zu erfassen und bei der Abfrage von normalisierten Types nur die Kanten aus einem einzelnen Normalisierungs-Verfahren zu nutzen.

Die Zeichennormalisierung ist letztlich nur einer der Problemkreise bei der Schaffung eines geeigneten Zugangs zu Vokabular und Textstellen. Andere Formen der Normalisierung, z. B. zur Abdeckung aller Schreibvarianten von Wörtern, können ebenfalls berücksichtigt werden. Wenn die normalisierten Formen einen sprachlich korrekten und lesbaren Wert besitzen müssen und sie für Analysen direkt verwendbar sein sollen, bietet es sich an, diese normalisierten Formen aufbauend auf der Zeichen-Normalisierung als eigenständige Wort-Annotation abzuspeichern. Dies ist etwa der Fall bei einer Normalisierung der Sprachstufe von alten in moderne Wortformen. Häufig existieren für diese Form der Normalisierung sowohl manuelle, als auch statistische und regelbasierte Verfahren²⁶, die – wie gerade schon angesprochen – in Kadmos gleichzeitig hinterlegt und wechselweise oder kombiniert verwendet werden können. Diese Form der Normalisierung ist

²⁶s. z. B. [BPD11] für die die Transformation vom Frühneuhochdeutschen zum Neuhochdeutschen

nicht an den Importvorgang gebunden, sondern kann auch zu einem späteren Zeitpunkt unter Berücksichtigung der im Graphen abgebildeten Kontexte automatisiert erfolgen, wie z. B. kürzlich in [Dem16] demonstriert wurde.

3.7 Flexibles graphbasiertes Information Retrieval

3.7.1 Retrievalverfahren und Textrepräsentation

Das Fachgebiet des Information Retrieval umfasst keine einzelne Technologie, sondern eine Vielzahl von Ansätzen, mit denen sich verwandte Problemstellungen lösen lassen. Baeza-Yates und Ribeiro-Neto umschreiben das Feld in [BYRN99] wie folgt:

Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest.

Manning, Raghavan und Schütze verwenden in [MRS08] eine kondensiertere Definition:

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Der Begriff Information Retrieval selbst geht auf einen Technologiereport im Umfeld des Bibliothekswesens aus den frühen 1950er Jahren zurück, s. [Moo51]. Seit dieser Zeit steht der Begriff in größerem Maße für die gewünschten Resultate der Verfahren als für konkrete Umsetzungsstrategien. Schon seit langer Zeit hat sich die Verschlagwortung, also das Vergeben von *Keywords*, als wichtiges Hilfsmittel zur Kategorisierung, Indizierung und damit „Auffindbarmachung“ von Dokumenten bewährt. Van Rijsbergen verweist in [Rij79] auf Arbeiten von Luhn, der in [Luh57] frequenzbasierte Auswertungen des Dokumentinhalts vornimmt, um darüber die Indizierung und das Auffinden von Dokumenten zu vereinfachen.

Grundsätzlich stellt sich die Frage, ob relevante Dokumente allein durch ihren textuellen Inhalt und ohne zusätzliche menschliche Kategorisierung gefunden werden können.

Schon in Arbeiten aus den 1980er und -90er Jahren wird festgestellt, dass das in einem Dokument verwendete Vokabular bereits eine erstaunlich gute Basis für automatische Indexsysteme bildet, welche einer manuellen Verschlagwortung (insbesondere, wenn sie dezentral von vielen Bearbeitern durchgeführt wird) nicht unterlegen ist. In [Lew92] wird dafür die Bezeichnung „*Equal Effectiveness Paradox*“ verwendet. Interessant ist noch ein zweiter in dieser Arbeit beschriebener Problemkreis, welcher als „*Perfect Query Paradox*“ eingeführt wird:

Für beinahe beliebige (thematisch abgrenzbare) Untermengen einer Dokumentenkollektion lässt sich eine stichwortbasierte Abfrage erstellen, die alle Dokumente dieser Untermenge (und kein anderes Dokument) zurückliefert. Als Begründung dafür wird angeführt, dass bereits ein individuelles Dokument üblicherweise über wenige sehr spezifische Wörter, die entweder gar nicht oder nur selten in anderen Dokumenten vorkommen, identifiziert werden kann²⁷. Eine Komposition solcher Keyword-„Signaturen“ ermöglicht die Konstruktion einer „perfekten“ Abfrage für alle als relevant erachteten Dokumente. Das Paradoxe daran ist, dass zum Finden einer perfekten Abfrage lexikalisch-inhaltliches Wissen über die gesamte Kollektion und speziell über die aufzufindenden Dokumente nötig ist – was voraussetzen würde, dass sie bereits bekannt sind.

In der Praxis wird dieser Widerspruch durch eine iterative Suchstrategie aufgelöst: Eine initiale, nicht perfekte Anfrage bietet Zugang zu einem Teil der Dokumentenkollektion, in dem sich neben irrelevanten Dokumenten auch „Zufallstreffer“ finden, aus denen sich dann weiteres zur Suchverfeinerung nutzbares Vokabular extrahieren lässt. Daraus kann abgeleitet werden, dass Interaktion und Exploration bereits bei konventionellen Recherchesystemen Teil des digitalen Recherche- und Forschungsprozesses sind, auch wenn diese Aspekte dort in der Regel nicht explizit erwähnt oder durch Werkzeuge und Bedienoberflächen unterstützt werden.

Aus der Erkenntnis, dass die Wortverwendung in den Dokumenten bereits einen ausreichenden Anhaltspunkt über deren inhaltliche Relevanz²⁸ darstellt, folgt direkt die Anwendbarkeit automatischer Methoden mit deutlich verringerten Anforderungen an eine manuelle Aufbereitung und Anreicherung der Korpora. Sie erlaubt die Erstellung universeller Verfahren auf der Basis von (im Allgemeinen sprachunabhängiger) Wortstatistik. Das Information Retrieval teilt sich daher auch viele Textmodelle und Repräsentations-

²⁷ Sehr spezifische und damit niederfrequente Terme gelten besonders auch in Anwendungsfällen der e-Humanities als geeignet für das Auffinden interessanter und bisher noch nicht erforschter Textstellen. Sie bilden damit wichtige „lokale Kontexte“.

²⁸ angesichts eines konkreten Informationsbedürfnisses

formen mit anderen Gebieten des Text Minings, wie sie in Abschnitt 2.2.4 auf Seite 39 eingeführt wurden.

Information Retrieval kann über die Betrachtung von Ähnlichkeiten von Wortansammlungen realisiert werden. Dazu können Methoden des semantischen Indizierens angewendet werden, um Dokumente nach ihrer Bedeutung gruppiert zu „verorten“. Dabei lassen sich die bereits angesprochenen Methoden zur Ermittlung latenter semantischer Dimensionen und Topics sowie *Word Embeddings* anwenden. Oftmals ist bei solchen Verfahren die Transformation großer Dokumente allerdings deutlich präziser möglich, als die der Anfrage, da diese meist sehr kurz ist und der zur Errechnung der semantischen Positionierung benötigte Kontext fehlt.

Für die Errechnung der Ähnlichkeit zwischen Dokumenten und Abfragen (als „Miniatur-Dokumenten“) können jedoch auch direkt die vorkommenden Terme betrachtet werden. Entsprechende Vergleichsvorschriften, wie das Cosinus-Maß, der Dice- oder der Jaccard-Koeffizient, s. z. B. [MS99], sind für die Vektorähnlichkeit im Vektorraum-Modell, wie es in [SWY75] beschrieben wurde, gut nutzbar. Dabei kann zusätzlich eine Termgewichtung stattfinden, etwa nach dem ebenfalls in [SWY75] vorgestellten Tf-idf-Maß.

Ein sehr populäres Verfahren zur Gewichtung des Termfrequenz-Teils von Tf-idf, das primär für *bag-of-words*-Ansätze verwendet wird, ist BM25, vgl. [RWJ⁺94]. Für BM25 existieren zahlreiche Erweiterungen und Parametrisierungen, die sich zum Teil aus (wenig kritisch hinterfragten) empirischen Optimierungen ergeben haben. Im einfachsten Fall errechnet sich die gewichtete Termfrequenz *wtf* laut der Veröffentlichung wie folgt:

$$wtf = \frac{tf^c}{\left(k_1 \left((1 - b) + b \frac{dl}{avdl}\right)\right)^c + tf^c}$$

mit *tf*, als der Termfrequenz im Dokument, *dl* als der Länge des aktuellen Dokuments in Wörtern, *avdl* als der durchschnittlichen Dokumentlänge im Korpus sowie einem freien Parameter *c*, der meist auf 1 oder als Funktion von *dl* und *avdl* festgelegt wird und *k₁*, einem weiteren freien Parameter. Diese gewichtete Termfrequenz wird mit der inversen Dokumentfrequenz (als $\log \frac{N+n+0,5}{n+0,5}$ mit *N* als der Dokumentanzahl und *n* als der Zahl der den Term enthaltenden Dokumente) multipliziert. Der so gebildete Wert wird über alle Suchterme aufaddiert als Ranking eines Dokuments verwendet.

In diesen Rechenvorschriften zeigt sich eine sehr globale Sicht auf das Korpus, für welche umfangreiche Vorberechnungs- und Indizierungsaufwände entstehen. Die durchschnittliche Dokumentlänge ist global, denn jedes hinzukommende Dokument ändert diese Größe und jede Abfrage dieses Wertes (ohne Vorberechnung) müsste alle Dokumente (und alle Wörter darin) betrachten. Die Größe n ist in begrenzterem Umfang global, da sie (je nach Worthäufigkeit) nur einen kleinen Ausschnitt des Korpus betrifft, und über eine termspezifische Zählvariable abgebildet werden kann, die nicht für jedes neue Dokument verändert werden muss. Lokal (bezogen auf das Dokument) ist nur die Termfrequenz. Es zeigt sich, dass eine solche Gewichtungsstrategie im Zusammenhang mit einer vorberechnungsfreien (und nur lokal effizient abfragbaren) Textrepräsentation in Graphdatenbanken nicht geeignet ist.

Grundsätzlich existieren auch gewichtungsfreie Verfahren. Eine der ältesten²⁹ Herangehensweisen an das Information Retrieval verwendet eine „logische Verknüpfung“ von Suchwörtern in einer Abfrage und wandelt diese in einfache Mengenoperationen der Ergebnismengen einzelner Stichwortanfragen um. Bereits Anfang der 1960er Jahre wiesen Verhoeff, Goffman, und Belzer in [VGB61] anhand einer mathematischen Formulierung des Retrieval-Prozesses nach, dass die (dort auch eingeführte) „Retrieval-Effizienz“ in diesem Szenario nicht optimal ist:

If the system responds to a request asking for "a" and "b" by giving the intersection of the responses it would have given to requests for "a" and "b" separately, it risks giving too much irrelevant material. If in response to a request for "a" or "b" it gives the union of the responses to the request for "a" and "b" if made separately, it risks leaving out relevant material. In both cases it will result in a decrease of efficiency.

Dennoch gibt es auch Gegenstimmen, die den nicht geradewegs zielführenden Kurs der booleschen Anfragen mit gegebenenfalls notwendigen Neuformulierungen der Anfrage als eine Chance für explorative Auseinandersetzung mit den Daten sehen, s. z. B. [HjØ15], wo es heißt:

Users may during the process explore how terms are used, how they co-occur and how knowledge is organised.

Prinzipiell ist festzuhalten, dass Abfragen initial nicht perfekt sind und nicht sein können und daher sukzessive erweitert werden müssen. Je mehr mögliche Schlagworte

²⁹und dabei heute noch (speziell auch bei Werkzeugen für die e-Humanities) gebräuchlichen

oder semantische Dimensionen während der Recherche dann als potentiell interessant identifiziert werden, umso mehr Dokumente, die (partielle) Treffer aufweisen (und im DH-Kontext nicht herausgefiltert werden können), werden zurückgeliefert. Dabei sollten Mengenoperationen auf den Ergebnismengen eher vermieden werden. An ihre Stelle kann ein geeignetes Ranking der Ergebnisse treten, welches keine Dokumente entfernt, aber potentiell wichtigere Dokumente priorisiert zurückliefert. Weitere Überlegungen zum (lokalen) Ranking werden noch in Abschnitt 3.7.4 auf Seite 108 angestellt.

Für den Rechercheprozess in den e-Humanities ergeben sich so grundsätzlich zwei Strategien, um in Dokumentkollektionen alle relevanten Informationen möglichst vollständig aufzufinden: Zum einen gilt es, das Korpus auf grundsätzlich interessante Teilbereiche einzuschränken, wofür sich eine Filterung anhand von Metadaten anbietet. Zum anderen soll die Suche nach Dokumenten stetig um zusätzliche relevante Aspekte erweitert werden. Die optimalen Ergebnisse erhält der Nutzer, wenn beide Strategien gleichzeitig verfolgt werden. In den folgenden Abschnitten werden die dafür benötigten Verfahren mit Bezug auf das Textdatenmodell vorgestellt.

3.7.2 Facettierung über nutzerspezifische Metadaten

Um die Nutzung von Suchfacetten zu motivieren, stellt Sacco in [Sac09] den universellen Wert eines explorativen Recherchevorganges heraus:

We contend that most "search" tasks are exploratory and imprecise in essence, and that using a focalized search paradigm in this context leads perforce to inadequate or frustrating user interactions.

Bei der Facettierung handelt es sich um das Identifizieren relevanter Inhaltskategorien und Metainformationen, nach denen die zu durchsuchenden Objekte gruppiert werden können. Dabei existieren kategoriale Facetten (z. B. der Name des Herstellers eines Produktes) und skalare Facetten (z. B. der Produktpreis). Die von den Objekten angenommenen Werte innerhalb dieser Facetten folgen einer bestimmten Verteilung, deren visuelle Darstellung dem Nutzer aggregierte Informationen (durchaus im Sinne des *Distant Reading*) über die Kollektion gibt.

Diese Facetten können darüber hinaus zur qualifizierten Festlegung von Filterungskriterien verwendet werden. In facettierten Oberflächen wird dann üblicherweise die Dar-

stellung der verbleibenden Facetten entsprechend an den neuen Ausschnitt angepasst. Dadurch lassen sich systematische Abweichungen zur Gesamtverteilung erkennen und so Hypothesen zu Korrelationen zwischen den Facetten aufstellen.

Durch die Nutzung des Graphenmodells lassen sich in Kadmos reichhaltige Filterungs- und Aggregierungsmöglichkeiten aus den Daten ableiten. Nicht nur direkt an den Dokumenten verankerte Metadaten, wie Genre und Datierung, können genutzt werden, sondern auch jede Art von durch Traversierung erreichbaren Kontexten. So können „Berufe der Großväter der Dokumentautoren“ eine durchaus denkbare (und abhängig von der Fragestellung eventuell sogar sinnvolle) Facette bilden – falls im System entsprechend auch biographische und prosopographische Daten in ausreichendem Umfang erfasst sind.

Wie z. B. in [HOH06] angedeutet ist diese Vielfalt an Möglichkeiten nur schwer allgemein zu handhaben:

Most facet browsers provide an interface to a single type of resource. Including multiple types, however, leads to an explosion in the number of corresponding properties and thus the number of available facets.

Daher wird in Kadmos keine generische Oberfläche zur facettierten Suche bereitgestellt. Entsprechende Funktionen sollten stets anwendungsfallspezifisch über spezialisierte Schnittstellen realisiert werden, wozu später in diesem Buch noch technische Details zur Sprache kommen.

Wenn in der explorativen Recherche ein so starker Fokus auf Metadatenfacetten gelegt wird, muss im Kontext der e-Humanities kritisch hinterfragt werden, ob diese den Forschungsprozess nicht über Gebühr beeinflussen. Stammen die Metadaten aus (potenziell intransparenter) externer Erfassung, so ist tatsächlich eine sorgfältige Prüfung auf Vollständigkeit, Korrektheit und semantische Kohärenz mit der Forschungsfrage erforderlich. Die dabei auffallenden Fehler können in Kadmos direkt durch entsprechende Datenbankbefehle behoben werden. Letztlich ist jedoch gerade im historischen Kontext ein Metadatum oft nicht eindeutig bestimmbar oder selbst Gegenstand kontroverser Theorien und laufender Forschung.

Um diesem Umstand gerecht zu werden, erlaubt das in Kadmos verwendete Datenmodell das Hinzufügen nutzerspezifischer Metadaten. In [Abbildung 3.8 auf der nächsten Seite](#) ist dargestellt, wie über Kanten mit dem Label `private` nutzerspezifische Änderungen

an den Metadaten des Korpus ausgedrückt werden können: Der Nutzer mit der ID 1 hat den Namen „Herodot“ eines Autors entfernt. Nutzer #2 hat die Autorenschaft dieses Autors zu einem Dokument entfernt und diesem stattdessen einen anderen Autoren zugewiesen. Nutzer #3 schließlich hat lediglich einen bestimmten Abschnitt mit diesem Autoren verknüpft.

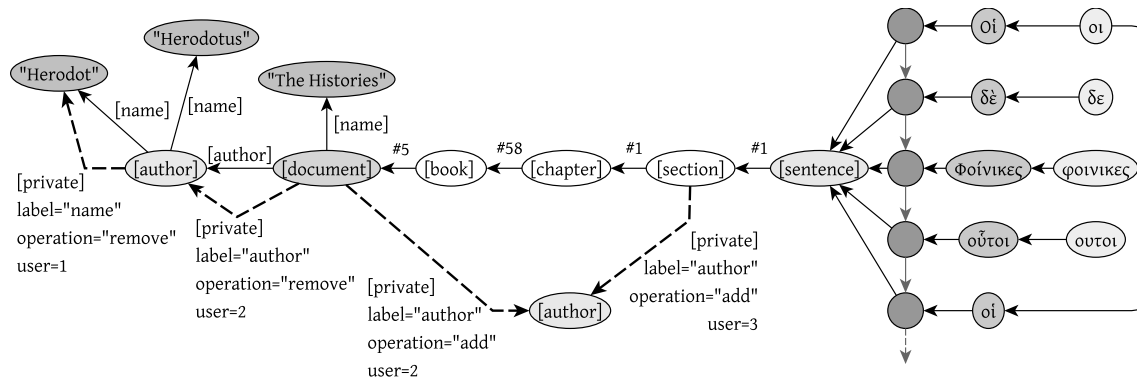


Abbildung 3.8: Schematische Übersicht nutzerspezifischer Metadaten

Sollen die nutzerspezifischen Angaben in einer Abfrage berücksichtigt werden, überlagern sie die Standard-Metadaten, falls die hinterlegte User-ID dem aktuell eingewählten Nutzer entspricht. Alle anderen Nutzer sind von dieser geänderten Sichtweise auf das Korpus nicht betroffen. Darüber hinaus ist auch denkbar, für diese Form der nutzerspezifischen Metadaten zusätzlich zu Userkennungen auch Gruppenkennungen zuzulassen, so dass die Angaben für mehrere Nutzer gelten können. Hierbei wäre noch zu klären, wie widersprüchliche Angaben aus mehreren Gruppen eines Nutzers zu handhaben sind.

Die Manipulationsmöglichkeiten sind dabei nicht auf Kanten beschränkt. Zum Ersetzen von Werten kann ein Knoten vom Typ proxy angelegt werden, in welchem der neue Property-Wert eingetragen ist. Der Originalknoten kann dann auf diesen Stellvertreterknoten über eine proxy-Kante, in welcher die aktuelle Nutzer-ID hinterlegt ist, verweisen. Anstatt solche Konstrukte direkt in den gemeinsamen Daten zu persistieren, ist generell auch denkbar, die nutzerspezifischen Teile über die sogenannte *Partition Strategy*³⁰ von TinkerPop zu isolieren.

³⁰http://tinkerpop.apache.org/docs/current/reference/#_partitionstrategy

3.7.3 Von Schlagwörtern zur konzeptbasierten Suche

Während (wie bereits beschrieben) das Auffinden relevanter Dokumente über „relevante“ Teile des genutzten Vokabulars möglich ist, stellt sich für automatische Verfahren im Information Retrieval die Frage, wie genau sich die Bedeutung einzelner Wörter auf die Dokumente überträgt und worin diese Bedeutung überhaupt besteht. In der Wortbedeutungslehre (Semasiologie) wird diese Thematik bereits seit langem untersucht und in diachroner Sichtweise im Rahmen der Begriffsgeschichte erforscht, s. z. B. [Fri06]. In konkreten und speziellen Forschungskontexten gehen diese Ansätze jedoch meist nicht weit genug und der Nutzer eines Recherchesystems muss sich die Bedeutung des Vokabulars (parallel zur Suche nach relevanten Dokumenten) selbst erschließen. Auf praktische Möglichkeiten dafür wird noch genauer in Abschnitt 3.8 auf Seite 109 eingegangen.

In konkreten Recheresituationen ist die Problemlage zunächst jedoch meist umgekehrt: Zu einer bestimmten Thematik sollen Dokumente gefunden werden, wofür Suchbegriffe benötigt werden, deren Bedeutung zur Thematik passt. Diese Suchbegriffe liegen jedoch (auch jenseits der strengen Sichtweise des *Perfect Query Paradox*) nicht vor. Piotrowski umschreibt diesen Umstand in [Pio12] für die historische Forschung wie folgt:

When searching a collection of historical documents, the situation is rather like searching in a foreign language: One may have some ideas about which words may be used, but one is not sure.

Der Nutzer hat also ein mentales Modell der zu untersuchenden Thematik und muss dieses als ein abgrenzbares „Konzept“ externalisieren und so mit geeigneten Begriffen unterfüttern, dass damit eine erfolgversprechende Abfrage des Korpus möglich wird. Ein Konzept ist dabei eine schwer zu definierende Einheit. Es wird hier weniger im Sinne einer „Konzeption“, also zielgerichteter Planung verwendet, als dass es als Grundgerüst für eine Ansammlung von „Bedeutung“ gesehen werden soll. Da diese Bedeutung zu großen Teilen mit sprachlichen Begrifflichkeiten umschrieben werden kann, zeigt sich hier eine Nähe zur „Konzeptualisierung“ im Umfeld von Ontologien, s. z. B. [Leh09]:

Eine Konzeptualisierung ist eine abstrakte und vereinfachte Sicht auf einen Teil der Welt, den man für einen bestimmten Zweck abbilden möchte. [Sie] enthält Konzepte (Vorstellungen über reale und ideelle Dinge) bzw. die Benennung dieser Konzepte durch Begriffe und die Beziehungen zwischen ihnen[...]

In Kadmos wird eine ebenso simple wie effektive technische Lösung für eine solche Konzeptualisierung vorgehalten. Bevor diese genauer beschrieben und anhand von Beispielen demonstriert wird, sollen zunächst einige hinführende theoretische Überlegungen angestellt werden. Neben den angesprochenen Arbeiten de Saussures zur Wortbedeutung auf syntagmatischer und paradigmatischer Ebene existieren dabei weitere grundsätzlich verwandte Vorarbeiten.

Zunächst kann auf die Theorie der Sprachlichen Felder (bzw. Wortfeldtheorie) zurückgegriffen werden. Wie in [Tri73] nachzulesen ist, motivierte beispielsweise Jost Trier bereits in den 1930er Jahren die Betrachtung von lokalen Wortkontexten zum Verständnis von Wortbedeutung, indem er feststellte:

In einem Gefüge hat alles nur Sinn aus dem Ganzen heraus. So stehn die Wörter einer Sprache nicht einzeln als Sinnträger da, sondern jedes Wort hat seinen Sinn nur daher, daß andere neben ihm Sinn haben.

Verwendungskontexte können genutzt werden, um Wörter nach ihren Bedeutungen zu gruppieren. In [Fri05] werden die Einflüsse des Ordnungsprinzips³¹ des 1933 erschienenen Übersichtswerks „Der deutsche Wortschatz nach Sachgruppen“ von Franz Dornseiff [Dor33] auf diesen Teilbereich der Semantik hervorgehoben. Dornseiff bezeichnet seine Begriffssammlung im Vorwort bereits u. a. als „Begriffsnetz“, was durchaus in passender Analogie zu modernen lexikalischen Ressourcen steht.

In [Hob70] wird dargelegt, wie die aus solchen Vorarbeiten weiterentwickelten, zunächst an formeller Logik orientierten Vorstellungen zur „Bedeutungsebene“ von Sprache nach und nach um psychologische Erklärungen ergänzt und langsam durch diese verdrängt wurden. In diesem Zusammenhang kamen Überlegungen zur sogenannten Framesemantik auf, vgl. etwa [Bus12]. Frames stellen dabei Einheiten von Weltwissen dar, in denen übliche Standardkontexte erfasst sind, und die Lücken aufweisen können, die mit konkreten Bezeichnern in einer zum Frame passenden Bedeutung gefüllt werden können. Ziem verbindet in [Zie05] die Framesemantik mit der sogenannten Diskursanalyse. Diese stellt einen Überbegriff für viele (nicht immer theoretisch und methodisch vollständig kompatible) Varianten der systematischen Beschäftigung mit Text in verschiedenen Zweigen der Sozial- und Geisteswissenschaften dar.

Diese Kette ist bei weitem nicht vollständig und soll nicht als die Beschreibung einer

³¹dort im inneren Titel als „synonymisch geordnet“ beschrieben

geradlinigen Entwicklung der Semantik-Forschung missverstanden werden. Dennoch zeigt sie, dass eine Konzeptdefinition im Sinne der Onomasiologie³² durchaus kohärent mit Theorien über Semantik und Wortbedeutung ist, und diese Gebiete darüber hinaus eine nicht geringe Verwandtschaft mit Analysemethoden der mit Kadmos adressierten Fachwissenschaften aufweisen.

Die praktische Anwendbarkeit dieser Überlegungen soll nun am Beispiel der Erweiterung von Suchvokabular bei der Stichwortsuche demonstriert werden. Dabei wird der Umgang mit Bedeutungsähnlichkeit hier am einfachsten Fall, nämlich an orthographischen Varianten verdeutlicht. Die Vorgehensweise ist jedoch unverändert auch für Synonyme und Begriffe gleicher „Sachgruppen“ anwendbar.

Abbildung 3.9 auf der nächsten Seite zeigt einen exemplarischen Recherchefall, in welchem quantitative Untersuchungen auf Basis der Artikeltexte und der Metadatenfacetten des Veröffentlichungsdatums aller Zeitungsberichte des NYT-Korpus durchgeführt werden. Dafür wird die Visualisierung einer „Calendar Heat Map“ gewählt, in welcher hier dunklere Kästchen eine höhere Anzahl pro Tag darstellen.³³ Für eine Recherche zu Alkoholkonsum bei Teenagern wird zunächst eine Suche nach Dokumenten vorgenommen, die sowohl den Begriff `teenagers` als auch den Begriff `alcohol` enthalten. Aus der Visualisierung ist eine Häufung dieser Dokumente ca. ab der zweiten Hälfte des Jahres 1999 ersichtlich.

Bei einer sorgfältigen Recherche wird dieses Ergebnis auf vielen Ebenen hinterfragt, u. a. auch durch Prüfung, ob ab diesem Zeitraum generell mehr über diese Bevölkerungsgruppe berichtet wurde. Die Suche nach `teenagers` offenbart dann eine sprunghafte Anhebung des täglichen Dokumentenvolumens für dieses Suchwort ab dem 1. November 1999. Ein solch starker, plötzlicher und dauerhafter Effekt ist entweder nur durch einen Fehler in der Software oder Datenquelle zu begründen (was hier nicht zutrifft) oder sie ist durch eine systematische Änderung in der Nutzung von Vokabular bei der Texterstellung erklärbar.³⁴ Letzteres kann gezeigt werden, wenn nach der (nun veralteten) Schreibweise `teen-agers` gesucht wird: Diese besitzt einen vollständig komplementären Zeitverlauf.

Mit dieser Erkenntnis lässt sich nun eine erweiterte Anfrage zusammenstellen, deren

³²als dem oben angesprochenen Gegenstück zur Semasiologie – zu diesem Feld und seinen Methoden s. z. B. [Grz11]

³³Für Vor- und Nachteile dieser Visualisierungstechnik s. z. B. [Yau13]

³⁴In Abbildung B/2 auf Seite 247 im Anhang sind zwei Beispiele für Histogramme gegeben, in denen tatsächliche Ereignisse eine sprunghafte Veränderung in der Zahl der Berichte zu einem Suchwort hervorrufen. Dem sofortigen Anstieg folgt dabei meist ein graduelles Abflachen.

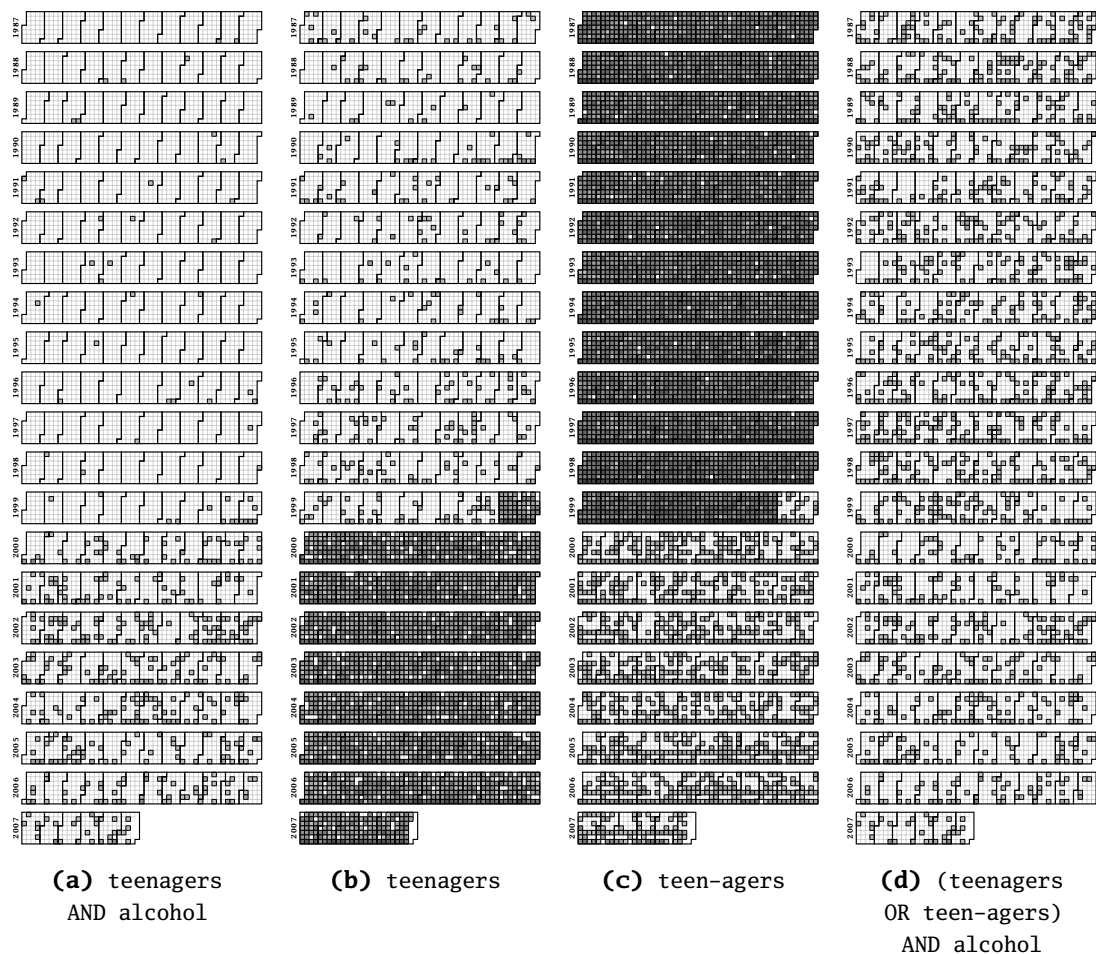


Abbildung 3.9: Jahreskalender-Histogramme (1987–2007) für die Zusammenfassung orthographischer Varianten in der Stichwortsuche des NYT-Korpus: (a) initiale Suche, (b) Analyse eines Einzelterms, (c) Auffinden einer Schreibvariante, (d) Kombinierte Suche mit beiden Varianten

Ergebnis keine klaren Hinweise auf quantitative Effekte über den abgedeckten Zeitraum hinweg mehr enthält. Für spätere Suchanfragen ist es wünschenswert, für Teenager immer beide Begriffe zu berücksichtigen. Zudem sollte auch die sprachliche Vielfalt abgebildet werden und z. B. für die Suche nach Aspekten des Alkoholismus stets auch Wörter wie liquor, booze und drunk abgefragt werden. Hier offenbart sich die direkte Anwendbarkeit einer onomasiologischer Konzeptdefinition auf den in den e-Humanities üblichen explorativen Recherchevorgang.

Für diese essenziellen Aufgaben der explorativen Suche wurde in Kadmos der Knoten-typ concept eingeführt. Jedes dieser Konzepte ist einem Nutzer zugewiesen und besitzt

eine durch diesen vergebene Bezeichnung. In Konzepten können andere Konzepte³⁵, normalisierte Types oder unnormalisierte Types (und prinzipiell sogar einzelne Token) zusammengefasst werden, indem ihnen diese über eine `belongs_to`-Kante zugewiesen werden. Beim Einfügen von Konzepten in andere Konzepte ist über eine geeignete Abfrage auszuschließen, dass dadurch direkte oder indirekte Zirkelreferenzen entstehen. Die Gesamtheit aller Konzeptknoten mit ihren entsprechenden Inklusionskanten besitzt somit immer die Topologie eines gerichteten azyklischen Graphen. Übergeordnete Konzepte „enthalten“ also andere in der Datenbank gespeicherte Elemente, weshalb sie – um ihre technische Natur hervorzuheben und sie vom Konzept als mentalem Modell abzugrenzen – auch als „Konzeptcontainer“ bezeichnet werden können.

In Abbildung 3.10 auf der nächsten Seite wird überblicksartig gezeigt, wie die vom Nutzer definierten Konzepte zusammengestellt werden können und wie sie Vokabular nutzen, welches im Graphen mittelbar mit Dokumenten verbunden ist, wodurch über sie Information Retrieval im klassischen Sinne möglich wird. Es wird dabei auch deutlich, wie die beiden Dokumente eine unterschiedliche thematische Abdeckung des gesuchten Konzepts aufweisen. Für eine Suche mittels Konzepten können je nach Anwendungsfall verschiedene Rankingkriterien aufgestellt werden, die Dokumente mit besonders breiter (oder aber in auffälligem Maße punktueller) Übereinstimmung mit den enthaltenen Subkonzepten bevorzugen.

Die gleichermaßen intuitive wie auch stark simplifizierte Konzeptmodellierung durch (hierarchisierte) Wortgruppierung, wie sie mit den vorgestellten Konzeptcontainern umgesetzt wurde, lässt sich nicht nur anhand der genannten theoretischen Vorarbeiten rechtfertigen, sondern auch durch ähnliche Ansätze aus der Praxis des Wissensmanagements stützen. Im Umfeld des Semantic Web wird unter dem Namen [Simple Knowledge Organization System \(SKOS\)](#) ein Vokabular für die Wissensorganisation verwaltet, welches unter anderem mit den Konstrukten „*Concept*“ und „*Collection*“ auf ganz ähnliche Weise arbeitet. Als nutzbare Relationen bietet [SKOS](#) u. a. die Prädikate „*broader*“ und „*narrower*“ (sowie explizit transitive Versionen davon), mit denen sich relative Granularitäten der Konzepte untereinander modellieren lassen. Auch in Kadmos sind viele weitere Ausprägungsformen von Relationen zwischen Konzepten denkbar und weitere Bedeutungsunterscheidungen für die `belongs_to`-Beziehung von Sub- zu Superkonzept leicht technisch umzusetzen. Die konkrete Semantik muss jedoch eng mit den jeweils

³⁵Konzepte des gleichen Nutzers oder mit einer dessen Nutzergruppen verbundene und damit „freigegebene“ Konzepte

Vokabular, dessen Annotationen oder die damit verknüpften nutzerdefinierten Konzepte zurückzuliefern und dabei anwendungsfallspezifisch zu gewichten. In Kadmos ist für diese Zwecke eine flexible Abfrage solcher „*Objects of Interest*“ möglich, wobei neben der Nutzung von Basisstrukturelementen auch beliebige künstliche Kontexte, wie Wortabstands-Fenster, satzzeichensensitive Fenster, vorkommende Muster von Annotationen, wie **Part-of-Speech (POS)**-Tags, uvm. definiert werden können. Nicht immer existiert für solche Konstrukte ein externalisierbares Referenzierungsschema. Jedoch können die internen Element-IDs zur Adressierung der jeweiligen Resultate über interaktive Oberflächen verwendet werden.

Zur Ermittlung einer Rangfolge für die Ergebnisse muss wieder einzelfallabhängig entschieden werden, welche Effekte dadurch erkannt, hervorgehoben oder eliminiert werden sollen. Die Bandbreite reicht von einer einfachen Zählung der *Traverser*, die nach der Anfrage auf den zurückgelieferten Knoten verweisen, bis hin zu komplexen Berechnungsvorschriften, welche auch globale Statistiken (oder Approximationen für diese) berücksichtigen. Es existieren bereits graphbasierte Maße für Relevanzberechnungen bei der Schlüsseltermextraktion [Bou13], der Termgewichtung [BL12] und der termbasierten Dokumentgewichtung [RV13], die als Ausgangspunkt für domänen- und fachfragenzentrierte Ansätze dienen können.

Der vorberechnungsfreie Zugriff auf die Datenbasis führt bei komplexeren Abfragen, die für solche Verfahren zum Teil nötig werden, zu Antwortzeiten jenseits einer wahrgenommenen direkten Interaktivität – der Vorgang benötigt also mehr als nur wenige Sekunden, bis die Ergebnisse vollständig vorliegen. Um dem Nutzer dennoch kein untätiges Warten zuzumuten, wird in Kadmos eine Retrievalstrategie bevorzugt, die an das in [FG06] eingeführte *Interrupt-driven Retrieval* angelehnt ist. Dieses lässt den Nutzer intermediäre Ergebnisse anfordern, welche den aktuellen, unvollständigen Abfragezustand abbilden, jedoch bereits ein Ranking der Ergebnisse vornehmen. In Kadmos besteht die Möglichkeit, solche Zwischenmeldungen periodisch liefern zu lassen, also die dafür notwendigen „Unterbrechungen“ des Retrievalvorgangs regelmäßig automatisch vorzunehmen. Durch die asynchrone Schnittstelle ist darüber hinaus die Umsetzung von Retrievalabfragen möglich, welche nicht extern unterbrochen werden müssen, sondern Zwischenergebnisse, sobald genügend neue Ergebnisse vorliegen, selbsttätig zurückmelden.

Ein so geschaffenes interaktives System, das Ergebnisse auf verschiedenen Granularitätsebenen und unter Berücksichtigung vieler Suchfacetten finden kann, erlaubt eine explorative Recherche, in welcher in kurzen Iterationsschritten der Recherchefokus an-

gepasst werden kann. Die vielen Möglichkeiten der fachfragengeleiteten und dennoch explorierenden Navigation durch das Korpus lassen die Recherche zu einem wertvollen und individuellen Vorgang der Wissenserschließung werden. Daher sollten die dabei erlangten Erkenntnisse und Entscheidungen möglichst explizit gemacht werden – zum späteren Nachschlagen, zur Projektdokumentation und als Kommunikationsmittel für den fachlichen Austausch.

In Kadmos ist die wichtigste Form der Dokumentation des Recherchevorgangs die Sammlung der durch den Nutzer angelegten Konzepte. Diese sind Artefakte, die exportiert, mitpubliziert, kritisch hinterfragt, verglichen, kommentiert usw. werden können. Der Diskurs um Interpretationen der Korpora und einzelner Themen kann damit in der Fachcommunity (falls gewünscht) auf expliziterer Ebene ablaufen.

Als Vannevar Bush im Jahr 1945 in einem visionären und vielbeachteten Essay das fiktive System „Memex“ beschrieb, mit dem sich große Datenmengen³⁶ speichern, explorieren und verknüpfen lassen, s. [Bus45], wies er bereits auf den größten Vorteil einer solchen Wissenssammlung hin: Das bewusste Verknüpfen von thematisch verwandten Informationsressourcen bei der Recherche in Form sogenannter *Trails* zur persönlichen Wissensorganisation. Eine ähnliche persistente Gruppierungsfunktion für Dokumente und Textstellen wird durch die Konzepte bereits jetzt geleistet. Eine Erweiterung dieser um Kommentare, den Verweis auf Strukturelemente und Annotationen und ggf. externe Wissensbasen könnte *Trails* auch in Kadmos realisierbar machen.

Wie angesprochen, lässt sich die so zu dokumentierende explorative Suche umso besser realisieren, je mehr geeignetes Suchvokabular bekannt ist. Neben der manuellen Analyse von Textstellen der Suchergebnisse lassen sich für diesen Zweck auch semi-automatische statistikbasierte und visuelle Methoden außerhalb des Information Retrieval nutzen, welche im nächsten Abschnitt zur Korpusexploration vorgestellt werden sollen.

3.8 Graphbasierte Korpusexploration

Die Erschließung von Vokabular ist ein wesentlicher Arbeitsschritt bei der Analyse von Korpora in den e-Humanities. Externe lexikalische Ressourcen bilden für spezifische Domänen und Forschungsfragen nur selten eine ausreichende Basis und auch mit Methoden

³⁶ „A library of a million volumes could be compressed into one end of a desk.“

des *Text Mining* können nicht alle relevanten Kontexte automatisch ermittelt werden. Deshalb werden explorative Zugänge zu den komplexen semi-strukturierten Datensätzen benötigt, für die sich eine enge Verzahnung mit Visualisierungsmöglichkeiten der *Visual Analytics* [KKEM10] anbietet. In letzter Zeit sind dort Verfahren für den Umgang mit Textdaten geschaffen worden, die meist auch unverändert in den e-Humanities anwendbar sind. Für die Anwendungsszenarien von *Visual Text Analytics* im Bereich der digitalen Geisteswissenschaften wird in [EAGJ⁺16] ein Vorgehensmodell skizziert, in welchem ein auf Forschungsfragen basierender Wissensaggregationsprozess in einzelne Aufgaben der Wissenserzeugung heruntergebrochen wird. Diese können durch geeignete visuelle und interaktive Werkzeuge zielgerichtet unterstützt werden.

Konkrete Visualisierungswerkzeuge für Textdaten sind zahlreich vorhanden, sie unterscheiden sich jedoch oft recht stark in ihrem genauen Fokus. Für die Darstellung und Kontrastierung von Wörtern in ihren jeweiligen Verwendungskontexten sollen hier als erstes die Einsatzmöglichkeiten von *Tag Pies*³⁷, s. [JBR⁺17], in Kadmos vorgestellt werden. Dabei handelt es sich um „Wortwolken“, die für mehrere Begriffe parallel erzeugt werden, wobei sie in Segmente eines „Kuchendiagramms“ eingepasst sind, welche die relativen Anteile der Suchwörter abbilden.

Abbildung 3.11 auf der nächsten Seite zeigt die Tag-Pies-Visualisierung für die normalisierten Types der Wortumgebungs-Kontexte dreier grammatikalischer Formen von „Κάδμος“. Dafür wurde die Layoutvariante „merged black“ gewählt, in welcher gemeinsame Kookkurrenten mehrerer angefragter Terme zusammengefasst in der Mitte dargestellt werden. Die Kopplung an Kadmos erlaubt, die Abfrage für ein frei wählbares Wortabstandsfenster (hier auf 5 Wörter festgelegt) vorzunehmen. Filterungen, wie sie für das Information Retrieval bereits vorgestellt wurden, sind zudem für diese Form der Korpusexploration analog umsetzbar.

Die Visualisierung gibt einen ersten Überblick, an den sich genauere philologische Betrachtungen anschließen müssen. Es sind dabei jedoch auch bereits für den Laien auffällige Effekte erkennbar. So befinden sich in den Kontexten der Nennung von Kadmos z. B. die Nominative der Namen seiner Frau (Ἀρμονία – Harmonia) und seiner Tochter (Σεμέλη – Semele) ausschließlich bei der Genitivform Κάδμου. Daraus könnte gefolgert werden, dass sie nur mit ihm im Text kookkurrieren, wenn sie als seine Verwandten vorgestellt werden. Zur Überprüfung solcher Ableitungen können die einzelnen Textstellen des gemeinsamen Auftretens interaktiv durch Klicken auf den entsprechenden Kook-

³⁷<http://www.tagpies.vizcovery.org/>

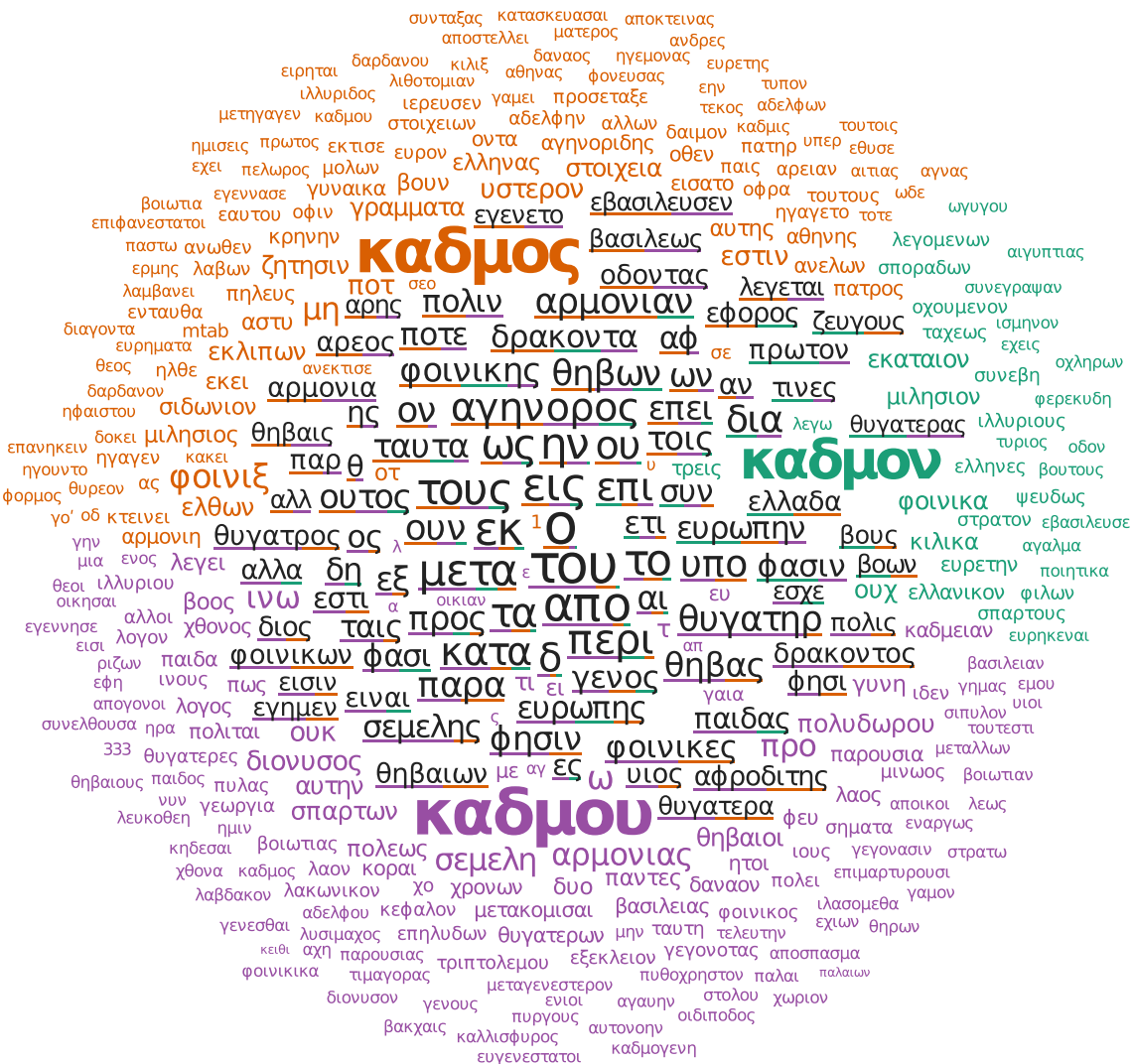


Abbildung 3.11: TagPies für den Vergleich grammatikalischer Formen des Eigennamens „Kadmos“ im **TLG-Korpus**

kurrenzterm aufgerufen werden. Für die Korpusexploration sind solche mehrstufigen Visualisierungstechniken (hier: Wortsuche, Kookkurrenzauswahl, Textstellenanalyse), die meist unter dem Begriff des *Focus+Context*³⁸ geführt werden, von großem Wert.

Verschiedene Teile des Vokabulars (z. B. die Wortarten) haben unterschiedliche Funktionen für die Kontexte und selbst „Stoppwörter“, also kurze häufige Wörter, weisen (meist aus grammatikalischen Gründen) eine besondere Affinität zu bestimmten Kontexten auf. Kurze Wörter lassen sich in der Darstellung gut überlesen, so dass in vielen Fällen auf ihre Filterung verzichtet werden kann und (im Sinne offener Forschung und auf formeller

³⁸eine etablierte Technik der Informationsvisualisierung, eingeführt in [SA82] als „Bifocal Display“

3.8 Graphbasierte Korpusexploration

Korrektheit bedachter Verfahren) verzichtet werden sollte. Stattdessen kann durch die Nutzung von Termgewichtung und Kookkurrenz-Signifikanzmaßen der Fokus potentiell auf seltenere Wörter gelegt werden. Dazu wird in Abbildung 3.12 eine Übersicht über den visuellen Effekt verschiedener Normierungsvarianten für die Termgewichte gegeben. Das Tf-idf-Maß wird dabei im Sinne von [SWY75] errechnet.

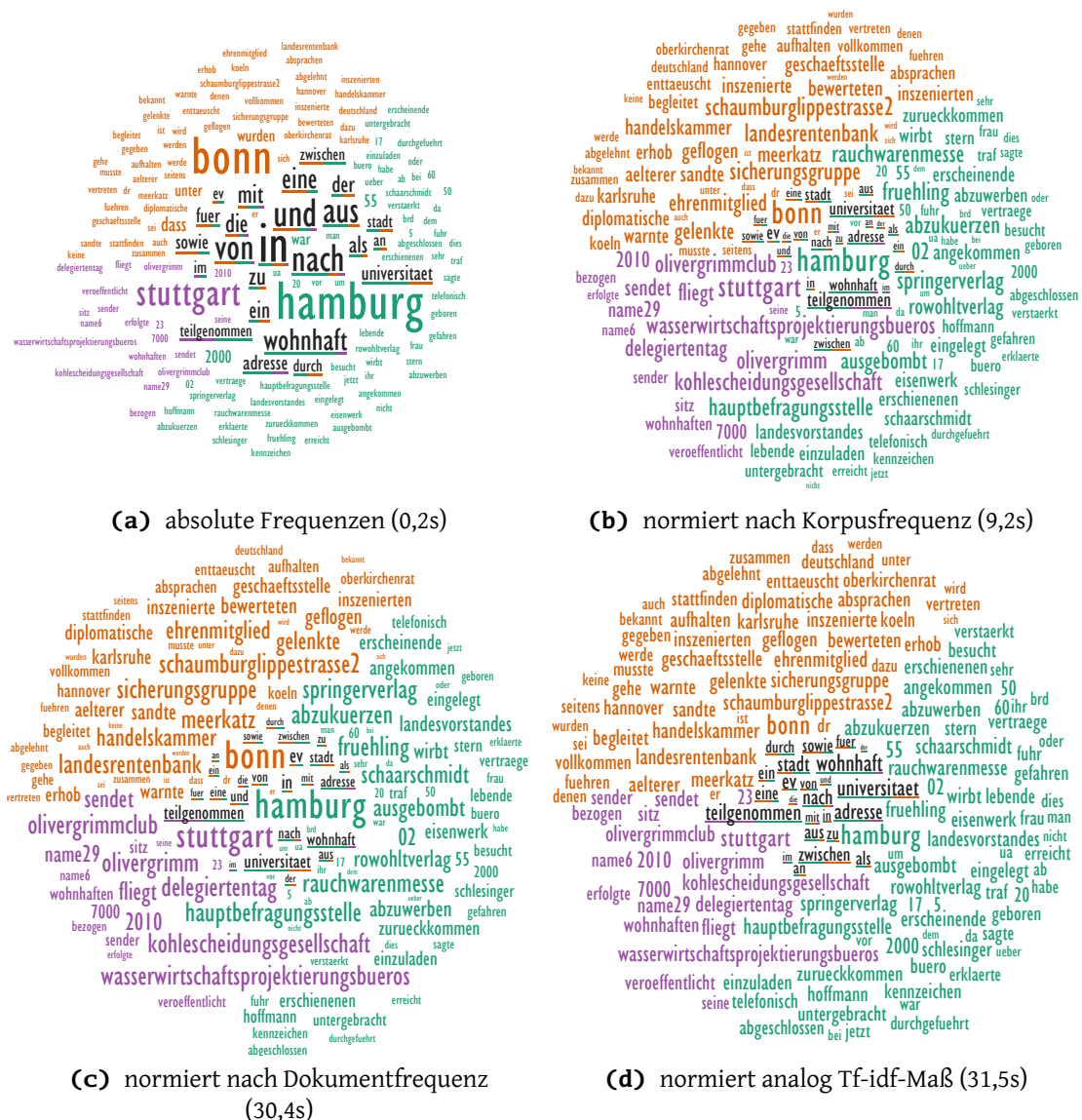


Abbildung 3.12: Kookkurrenzterme in direkter Nachbarschaft der Wörter bonn, hamburg und stuttgart im STASI-Korpus visualisiert mit verschiedenen Gewichtungsmethoden (incl. der jeweiligen Abfragezeiten ohne Vorberechnung)

Es lässt sich ablesen, dass seltenere Wörter, wie z. B. „Kohlescheidungsgesellschaft“ durch die Normierung stärker hervortreten. Solche Effekte sind in der historischen Forschung

oft gewünscht, in anderen Feldern, wie der Stilometrie, wo eine Kontrastierung von Vokabularnutzungsweisen zwischen Autoren stattfindet, werden dagegen meist systematische Abweichungen im Bereich häufiger Wörter untersucht.

Während in Tag Pies für die Wortkontexte ein Bag-of-Words-Ansatz gewählt wird, kann für andere Visualisierungsansätze die in Kadmos gespeicherte Wortreihenfolge genutzt werden. Neben der Erstellung klassischer Konkordanzlisten können die Umgebungen auch aggregiert angezeigt werden. Abbildung 3.13 zeigt ein Beispiel für die häufigkeitssortierte Auflistung von linken und rechten Nachbarn für mehrere Types, die dem gleichen normalisierten Type zugeordnet sind. Die Visualisierungskomponente für das Werkzeug Netspeak³⁹, s. [RGP⁺12], unterstützt darüber hinaus eine Aggregation von Wortkontexten bis zur Länge 5, die beliebigen Mustern folgen können, also z. B. den Nutzer mögliche Ersetzungen von Teilen fixer Phrasen finden lassen. Auch eine solche Oberfläche könnte mit Kadmos als Backend betrieben werden. Eine Längenbeschränkung wäre dabei technisch nicht notwendig. Allerdings wäre die Abfrageperformance der Graphdatenbank für die interaktive Aggregation häufiger Muster ggf. nicht ausreichend.

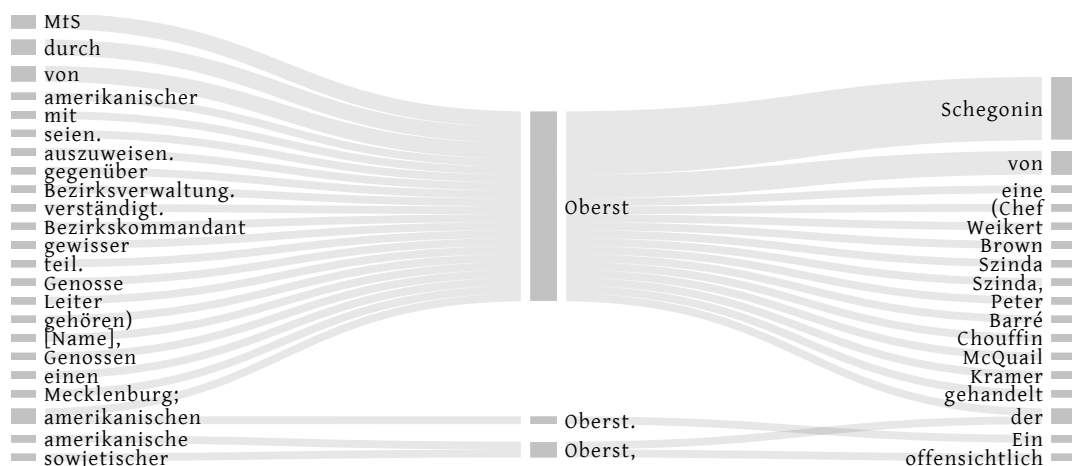


Abbildung 3.13: Visualisierung aggregierter Konkordanzdaten für direkte Nachbarn des unnormalisierten Types „oberst“ im STASI-Korpus

Für diese Art von Kontexten existieren weitere Visualisierungsmöglichkeiten, etwa der in [DK15] vorgestellte „WordWanderer“, der als „casual form of engagement with text“ unter Nutzung von Wortwolken und Kollokationsfenstern der fixen Breite 7 eine Korpusexploration ermöglicht. Das Programm verfügt über kein Datenbankbackend, und kann daher nur mit Einzelkorpora limitierter Größe genutzt werden: „At present, WordWanderer only uses individual texts of up to about 50,000 words [...], but it can be extended to much larger corpora

³⁹<http://www.netspeak.org/>

when coupled with a backend and a database.“ – Auch hier bietet sich Kadmos perspektivisch als Backend an.

Einen ähnlichen Ansatz wie die Metadatenfacettierung in Kadmos verfolgt das Werkzeug DiaCollo, s. [JGW16]. Dort werden für die Kollokationsanalyse und -darstellung auch Dokumentenzeitstempel berücksichtigt, so dass eine diachrone Sicht auf kookkurrierende Kontextwörter möglich wird. Die Breite der auszuwertenden Nachbarschaftsfenster kann dort allerdings nicht zur Laufzeit eingestellt werden, da diese Justierungen eine Neuerstellung der Datenbasis erfordern.

Kadmos ist also kompatibel zu bestehenden Werkzeugen für die Korpusexploration und das darin verwendete graphförmige Textdatenmodell bietet zudem viele Möglichkeiten zur Bereitstellung von Anfrageergebnissen in etablierten Formaten. Um die existierenden Kontexte sichtbar zu machen, kann zudem eine Graphenvisualisierung für Ausschnitte der Datenbank vorgenommen werden. Dafür wurde das interaktive Werkzeug „Kadmos Navigator“ integriert. Es erlaubt eine Visualisierung der Zusammenhänge im Graphen und unterstützt das Nachladen weiterer Teile in die bestehende Ansicht durch Interaktion mit den dargestellten Knoten. Für die visuelle Repräsentation wird der übliche Weg über eine Planarisierung des Graphen vorgenommen. Einen Überblick über die dabei grundsätzlich relevanten visuellen Parameter gibt [Pur02]. In der konkreten Umsetzung wurde der kräftebasierte Layoutalgorithmus ForceAtlas2, s. [JVHB14], verwendet und die Visualisierung über die Javascript-Bibliothek Sigma⁴⁰ umgesetzt. In Abbildung B/3 auf Seite 248 im Anhang ist ein darüber dargestellter Ausschnitt der Datenbank zu sehen.

All diese vorgestellten Explorationsmöglichkeiten bedienen sich in Kadmos keiner Vorberechnung und keiner anderen Unterstützungsstruktur als der des Graphen (und dessen Indexsystems). Entsprechend ist die Betrachtung der Leistungsfähigkeit des Systems als Teil der nun folgenden Evaluierung eine wesentliche Voraussetzung für seine breite Anwendbarkeit.

3.9 Evaluierung

Kadmos ist ein anpassbares Rechtersystem, mit dem sich viele Aufgaben der automatischen Sprachverarbeitung auf eine neuartige Weise bearbeiten lassen. An dieser Stelle

⁴⁰<http://sigmajs.org/>

sollen die Leistungsgrenzen des Systems ausgelotet werden. Eine Evaluierung der „Qualität“ des Systems ist dagegen nur schwer zu erbringen, da die Zweckmäßigkeit sich nur anhand eines bestimmten Einsatzzwecks beurteilen lässt. Zum Verhalten der Technologie in konkreten Einsatzszenarien werden im nächsten Kapitel noch weitere Betrachtungen angestellt. Hier sollen zunächst nur die Ausführungsparameter des Systems untersucht werden.

In Abbildung 3.14 auf der nächsten Seite ist die Anfragezeit für die Traversierung der lokalen Kontexte von Wörtern abgebildet. Die Messung erfolgte auf dem kompletten STASI-Korpus. Dabei wurde für alle 80.584 normalisierten Types eine Abfrage ausgeführt, die über die Type-Knoten die zugeordneten Token anläuft, ihre direkten linken und rechten Nachbarn traversiert und schließlich deren (wiederum indirekt über unnormalisierte Types) verknüpfte normalisierte Token erreicht. Diese Endpositionen werden (um ein realistisches Anfrageszenario nachzubilden) anschließend gruppiert, innerhalb der Gruppe gezählt und nach dieser Anzahl absteigend sortiert. Zusätzlich wird für die Gruppe der gespeicherte Textwert des Knotens ausgelesen. Die Anfragen erfolgten sequenziell in zufälliger Reihenfolge in einem einzigen Durchgang und ohne Parallelisierung (*single-threaded*). Das System wurde zuvor neu gestartet, ein „Vorwärmen“ des Zwischenspeichers fand also nicht statt, was die gelegentlichen hohen Ausschläge des Maximalwerts erklärt.

Für Wörter mit bis zu 10.000 Nachbarn kann die Anfrage in der Regel in weniger als einer Sekunde beantwortet werden. Für niederfrequentes Vokabular lässt sich eine Antwortzeit von wenigen Millisekunden realisieren. Im Diagramm lässt sich gut ein linearer Zusammenhang zwischen Anfragezeit und Ergebnisgröße ablesen. Es handelt sich also um einen „outputlinearen“ Vorgang. Das damit verbundene Systemverhalten im Hinblick auf Wartezeiten kann Nutzern in der Regel gut kommuniziert werden und gibt bereits während der Ausführung der Anfrage ein Feedback zur erwartbaren Ergebnismenge: Wenn die Anfrage lange dauert, werden wahrscheinlich viele Ergebnisse zurückgeliefert. Das ermöglicht ggf. den frühen Abbruch einer thematisch deutlich zu breit gefassten Anfrage.

Generell dürften die Antwortzeiten jedoch angemessen ausfallen. Abbildung 3.15 auf Seite 117 zeigt, dass im Text die meisten Wörter nur sehr selten vorkommen und das häufige Vorkommen eines Wortes sehr selten ist. Mehrere zehntausend normalisierte Types kommen im STASI-Korpus nur ein einziges mal vor und jeweils nur einzelne tauchen im Fließtext mehr als zehntausend mal auf. Diese prinzipielle Häufigkeitsverteilung des

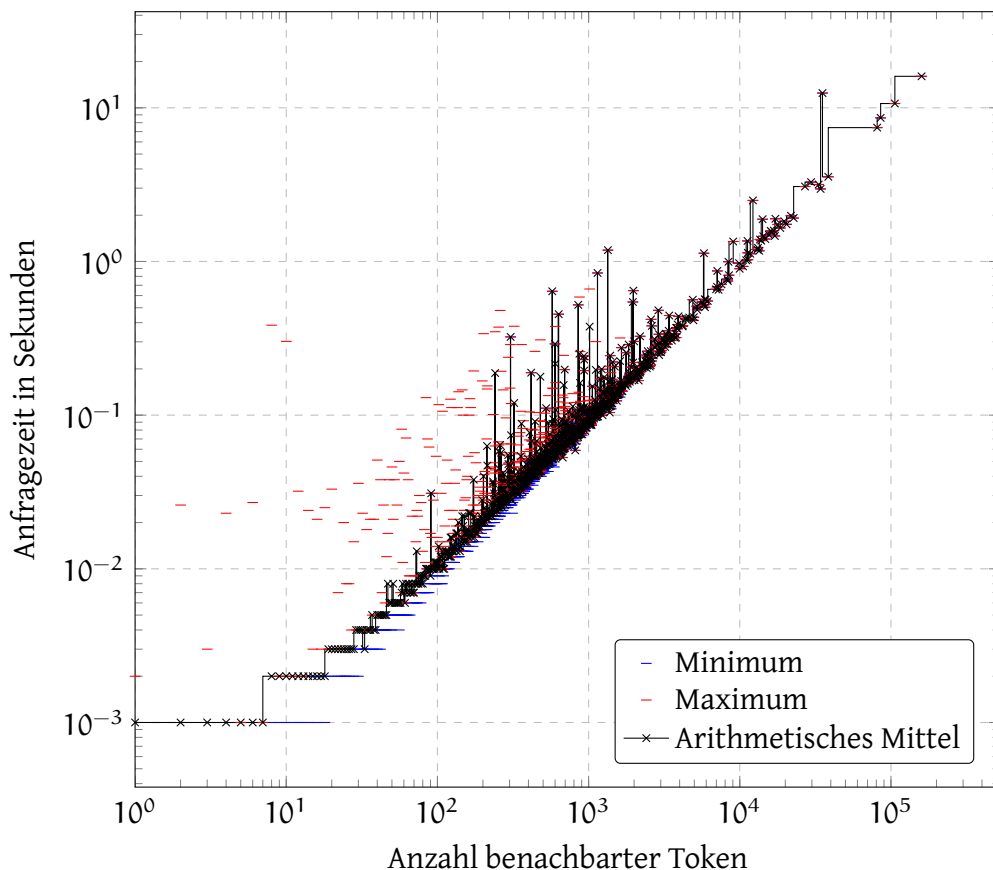


Abbildung 3.14: Abfragezeit für gruppierte normalisierte Kookkurrenzwörter im [STASI-Korpus](#), ermittelt auf einer frisch gestarteten Kadmos-Instanz mit genügend großem Arbeitsspeicher

Vokabulars von Texten ist gut erforscht und wurde von George Kingsley Zipf bereits in den 1930er Jahren (anhand von nach Rang sortierten Wortfrequenzen) mathematisch beschrieben, s. u. a. [\[Zip32\]](#).

In der Praxis bedeutet das für die Evaluierung der Abfragegeschwindigkeit, dass lokale Kontexte von Wörtern in der überwiegenden Anzahl von Fällen sehr schnell ermittelt werden können. Die Abfrage von Konzepten, die viele Wörter umfassen, ist ebenfalls schnell, so lange die Wörter sehr spezifisch sind. Ein Konzept, das hochfrequente Begriffe (z. B. „Stoppwörter“, wie Artikel und Präpositionen) gruppiert, kann trotz geringem Vokabularumfang sehr lange Abfragezeiten verursachen. Für solche omnipräsenten Terme erweitert sich die durch Graphdatenbanken unterstützte Sicht auf lokale Kontexte zu einer globalen Betrachtung des kompletten Textes. Hier können vorberechnungsfreie Statistiken nicht interaktiv abgefragt werden und auch Kontextabfragen übersteigen die von herkömmlichen Volltextsuchsystemen gewohnten Antwortzeiten um ein Vielfaches.


```

1  ### Datenbankabfragen
2  raw_results = @corpus.find_normalized_type_nodes.map do |nt|
3    start_time = Time.now
4    neighbors = nt.in(:normalization).out(:type_token).both(:next).in(:type_token
5      ).out(:normalization)
6    groups = neighbors.map{|rnt|rnt[:string_value]}.group_count
7    duration = Time.now - start_time
8    hits = groups.values.inject(&:+)
9    next [hits, duration]
10 end
11 ### Ergebnisstatistik
12 results = raw_results.group_by(&:first).map do |count, entries|
13   times = entries.map(&:last)
14   average = (times.inject(&:+)*1.0/times.count)
15   next [count, [average, times.min, times.max]]
16 end
17 ### Ausgabe als Tab-separierte Liste
18 out.object results.sort_by(&:first).map{|l|l.join("\t")}.join("\n")

```

Quelltext 3.3: Benchmarking-Code für gruppierte normalisierte Kookkurrenzwörter

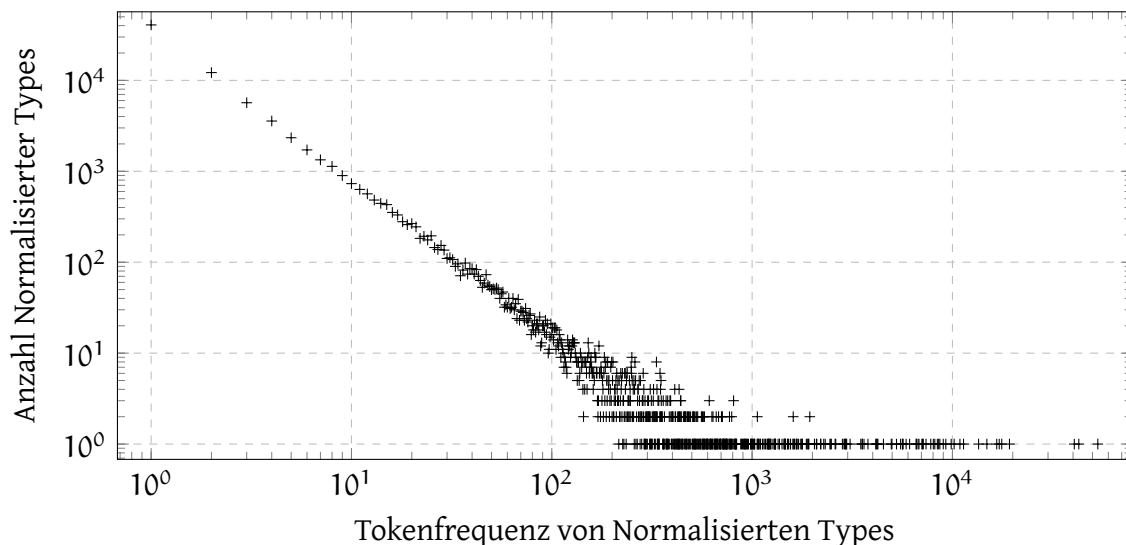


Abbildung 3.15: Normalisierte Types nach Tokenanzahl im [STASI-Korpus](#)

Nicht alle Abfragearten weisen ein lineares Antwortverhalten bezüglich Eingabe- oder Ausgabegröße auf. Neben der grundsätzlichen Analyse der algorithmischen Komplexität der Abfragen sind weitere Zusammenhänge zu berücksichtigen, wenn die Ausführungsaufwände für Abfragen praxisnah bestimmt werden sollen: Zum einen sind die Traversierungsvorgänge grundsätzlich von der Topologie des Graphen abhängig. Zwar

ist das Schema für die Textrepräsentation einheitlich, jedoch bestimmen die Sprache und Textgattung der repräsentierten Dokumente sowie die dokumentübergreifenden Verknüpfungen (z. B. über Metadaten oder Annotationen) wesentlich den Grad der Verknüpfung in der Datenbasis. Zudem ist oft nicht abzuschätzen, inwieweit technische Parameter, wie Effekte der Speicherverwaltung, und der Parallelisierbarkeit, bei verschiedenen *Workloads* ins Gewicht fallen. Komplexitätsbetrachtungen auf rein theoretischer Ebene – etwa anhand der expliziten Traversierungsvorschriften – sind daher nicht aussagekräftig für das reale Verhalten des Systems. Die tatsächlichen Ausführungszeiten können jedoch systematisch erfasst werden und dann durch geeignete Schätzverfahren im Kontinuum zwischen den bekannten Komplexitätsklassen positioniert werden. Ein geeignetes Werkzeug für diese Form der Komplexitätsbestimmung stellen die in [TWD⁺13] vorgestellten *Complexity Plots* dar.

Als nächstes soll sich eine Untersuchung der Skalierbarkeit des Systems anschließen. Dabei steht nicht die Skalierung der externen verteilten Speicher-Backends im Fokus. Diese wurden, wie eingangs in diesem Kapitel beschrieben, bereits andernorts untersucht und spiegeln hauptsächlich technische Entscheidungen und Kompromisse innerhalb des vom CAP-Theorem abgesteckten Rahmens wider.

Stattdessen wird hier die Parallelisierbarkeit der lokalen Abfrageausführung mit dem lokalen Backend untersucht. In Abbildung 3.16 auf der nächsten Seite kann der Geschwindigkeitszugewinn bei Erhöhung der Threadanzahl bei nebenläufig abgearbeiteter Graphentraversierung abgelesen werden. Die Abfrage ist dabei die selbe wie im vorangehenden Beispiel: das Finden direkter Nachbarschafts-Kookkurrenten. Hier wurden jedoch nur Wörter betrachtet, deren Häufigkeit deutlich größer als die Thread-Anzahl ist, um zu häufige Kontextwechsel zu vermeiden.

Die sublineare Beschleunigung bei zwei Threads lässt sich eventuell mit zusätzlichen Aufwänden für die Verwaltung von Threadsicherheits- und Threadkommunikations-Mechanismen erklären. Die superlineare Beschleunigung von vier bis ca. zehn Threads könnte mit der besseren Handhabbarkeit von Latenzen des Festplattenzugriffs erklärbar sein, verglichen mit einer rein sequentiellen Abfrage verschiedener Bereiche des Graphen. Die stagnierenden und später sogar abfallenden Beschleunigungswerte deuten auf eine Erhöhung der Koordinationsaufwände für Threads und größere gegenseitige Abhängigkeiten hin. Eingehendere Untersuchungen dazu wurden aber nicht angestellt. Hier verbirgt sich ggf. noch Optimierungspotential. Es zeigt sich aber, dass eine zehnfache Beschleunigung einer üblichen Abfrage durch *Multithreading* durchaus realistisch ist.

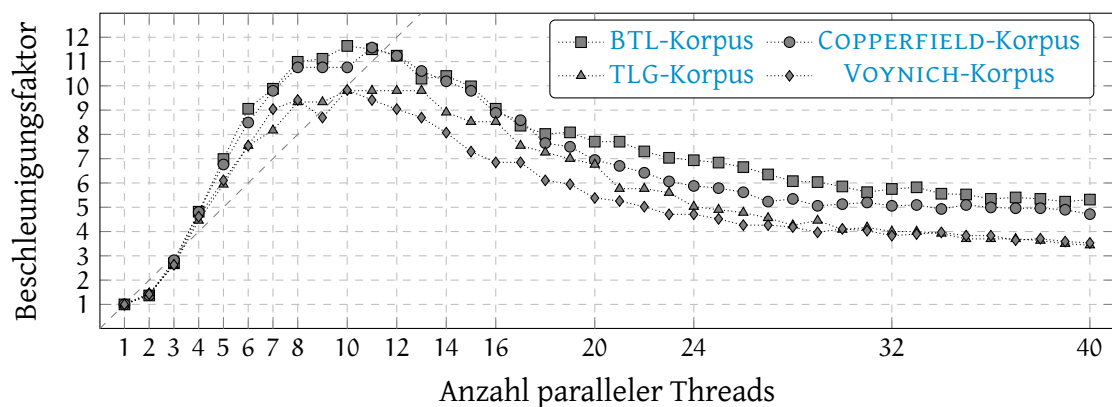


Abbildung 3.16: Median der Abfragebeschleunigung durch parallelisierte Traversierung auf einem System mit vier Achtkern-Prozessoren, gebildet über je 100 Durchgänge

Was in der Grafik zusätzlich deutlich wird, ist die Abhängigkeit der Beschleunigung vom Korpus. Denkbare Faktoren dafür sind Sprache und Genre der Texte und die davon abhängigen Besonderheiten in der Vernetzung des Graphen. Die Größe des Korpus scheint auf die Parallelisierbarkeit nur einen geringeren Einfluss zu haben⁴¹.

Im nächsten Schritt soll eine Untersuchung des Arbeitsspeicher-Bedarfs des Kadmos-Systems folgen. Dabei wird es in typischer Konfiguration, befüllt mit einigen Korpora (darunter auch größere), unter Nutzung lokaler Speicher-Backends betrachtet. Der maximal nutzbare Arbeitsspeicher wird über den Parameter `Xmx` der Java-Laufzeitumgebung reguliert.

In Tabelle 3.3 auf der nächsten Seite wird das Verhalten des Systems unter beschränkter Zuweisung von Arbeitsspeicher untersucht. Der dort aufgeführte Festplattenplatz pro Korpus bezieht sich auf die persistierten Daten der Berkeley DB zuzüglich der Dateien des lokalen Lucene-Volltextindex. Die angegebenen Zeiten beziehen sich auf das Sortieren aller Types nach ihrer jeweiligen Tokenanzahl unter Angabe des jeweiligen String-Werts. Die Abfrage erfolgt in einem einzelnen Thread.

Im Hinblick auf die Speicheranforderungen lässt sich feststellen, dass das System (ohne weitere dahingehende Optimierungen) zwar für den Einsatz auf Desktop- oder Serverumgebungen, nicht aber nicht für den Betrieb im *Embedded*-Bereich geeignet ist.

Im Rahmen der Leistungsevaluierung von Kadmos-Services bleiben noch einige Bemerkungen zum *Caching* zu tätigen: Bei der Anfrage kann es sinnvoll sein, in lokalen Variab-

⁴¹Das BTL-Korpus und das TLG-Korpus sind deutlich größer als die beiden anderen

-Xmx	Fehlermeldung beim Startvorgang bis 256 MB:		
1m			
2m, 4m, 8m			
16m, 32m			
64m, 128m			
256m			
-Xmx	VOYNICH-Korpus	REUTERS-Korpus	BTL-Korpus
512m	15s	—, 	—, 
1g	5s	147s, 	—, 
2g	5s	41s	—, 
4g	4s	40s	3790s, 
8g	5s	39s	439s
16g	5s	40s	265s
32g	5s	41s	275s
64g	5s	42s	283s
128g	4s	43s	296s
256g	5s	45s	298s
Top-Type:	661 × „  “	119584 × „the“	356144 × „et“
Festplattenplatz:	50 MB	2621 MB	17819 MB

Legende für Fehlermeldungen:

-  Error occurred during initialization of VM
-  OutOfMemoryError: Java heap space
-  OutOfMemoryError: GC overhead limit exceeded
-  Could not instantiate implementation: BerkeleyJESToreManager
-  TitanException: Could not execute operation due to backend exception

Tabelle 3.3: Antwortzeiten und Fehlermeldungen bei der Ausführung von Kadmos mit beschränktem Arbeitsspeicher. Bis 256 MB ließ sich das System nicht starten.

len Zwischenergebnisse vorzuhalten (etwa über *hashes* mit entsprechenden eindeutigen Schlüsseln), die dann in der weiteren Anfrage (ohne eine redundante Berechnung zu erfordern) genutzt werden können.

Darüber hinaus kann darüber nachgedacht werden, häufig genutzte, oder weitestgehend statische Kennzahlen (z. B. die Zahl ausgehender Kanten eines Typs) direkt im Property Graph zwischenzuspeichern und so eine abfrageübergreifende Beschleunigung zu erzeugen. Entsprechend müssen bei dynamisch veränderlichen Werten Strategien zur Invalidierung gefunden werden (Neuberechnung bei zu alten Zeitstempeln, Neuberechnung beim Hinzufügen neuer Knoten oder Kanten in der Nachbarschaft). Hier bieten sich unterschiedliche Herangehensweisen bei abgeschlossenen und veränderlichen, offenen Korpora an. Die tatsächliche Wirksamkeit muss dabei auch immer in Kombination mit dem jeweils gewählten Speicher-Backend evaluiert werden.

Auch wenn hier, wie eingangs erwähnt, keine Evaluierung der Zweckmäßigkeit oder gar Qualität von speziellen Verfahren durchgeführt werden kann, so soll doch darauf hingewiesen werden, dass Kadmos grundsätzlich zur Durchführung solcher Evaluierungen geeignet ist. Gerade weil z. B. das vorgestellte Feld des Information Retrieval im Rahmen der e-Humanities stets eine aufgabenspezifische Anwendung von spezialisierten Algorithmen mit sich bringt, sind für eine Evaluierung auf diesem Gebiet stets recht wenig vergleichbare Ansätze verfügbar.⁴² Vergleiche können zudem sinnvollerweise nur anhand gleicher Datengrundlagen und mit gleichen Testbedingungen erfolgen.

Für solche gezielten Tests empfiehlt sich eine Vergleichsinfrastruktur, wie z. B. das „Testbed for Information Retrieval Algorithms“ [GSBH12], welche grundsätzlich auch in Kadmos integriert werden kann. Die allgemein angebotenen Möglichkeiten zur Erweiterung des Systems werden nun in den folgenden Abschnitten vorgestellt.

⁴²Die isolierte Betrachtung einzelner Verfahren ist insbesondere dann nicht zielführend, wenn jeweils nur mit einem schwachen *Baseline*-Verfahren verglichen wird, wie [AMWZ09] argumentiert. Daraus resultieren oft schlecht verallgemeinerbare und kombinierbare Optimierungen, die sich nicht in anderen Kontexten bewähren.

3.10 Erweiterungsmöglichkeiten

3.10.1 Anlegen neuer Service-Endpunkte

Entsprechend des in [3.1 auf Seite 74](#) beschriebenen prototypzentrischen Ansatzes für Kadmos wird ein sehr flexibles System für die Erstellung und Änderung von Schnittstellen zur Nutzung der Backend-Funktionalitäten benötigt. Es existiert nur eine begrenzte Zahl fest vordefinierter [API](#)-Endpunkte. Anpassungen und Neuentwicklungen an den Services sind übliche Arbeitsschritte in der Verwendung der Software in konkreten Projektkontexten. Sie sollten ohne Neustarts und Wartezeiten im laufenden Betrieb durchgeführt werden können. Ruby als interpretierte Sprache bietet ideale Voraussetzungen für einen solchen Arbeitsmodus.

Die [API](#)-Endpunkte werden über Ruby-Quelltextdateien im Ordner `controllers` definiert. Der Dateinamenpräfix `api_` unterscheidet sie von regulären *Controller*-Dateien, die [API](#)-unabhängige Funktionen der Webapplikation verwalten. Im Folgenden wird demonstriert, wie einfach eine zusätzliche Schnittstelle zu Kadmos hinzugefügt werden kann. Als kompakter Beispiel-Anwendungsfall wird hier der Export von lokalen Umgebungen für mehrere normalisierte Token als Datei in einem Graphen-Austauschformat vorgestellt. Eine solche Datei kann z. B. für die Weiterverarbeitung in externen Graphanalyse-Werkzeugen wie Gephi verwendet werden.

Nach dem Anlegen einer neuen Datei `controllers/api_neighborhoods.rb` ist es möglich, den darin enthaltenen Programmcode mit einem [API](#)-Aufruf unter der Adresse <http://localhost/api/neighborhoods> ausführen zu lassen. Soll nicht das Hauptkorpus verwendet werden, kann die Abfrage um die Parameterangabe `?corpus=Datenbankname` erweitert werden. Das entsprechende mit Pacer abfragbare Korpus-Objekt steht im Quelltext der [API](#)-Datei dann unter dem Variablennamen `@corpus` zur Verfügung. Für die Ausgabe von Werten stehen mehrere Varianten zur Verfügung. Während die Methode `out.data` beliebige Zeichenketten in den Ausgabestrom schreibt, sendet `out.object` eine [JSON](#)-kodierte Repräsentation ihres Aufruf-Arguments an den Client.

Quelltext [3.4 auf der nächsten Seite](#) zeigt eine mögliche Implementierung der im oben beschriebenen Beispiel gewünschten Funktionalität: Die gewünschten kommaseparierten Wörter werden aus dem Parameter-Objekt ausgelesen, welches in Sinatra einen Zugriff sowohl auf Parameter in der [URL](#) als auch auf im [HTTP](#)-Request-Body mitgesendete

„Post“-Parameter zulässt. Diese Wörter werden anschließend in einer gemeinsamen Index-Abfrage⁴³ zum Auffinden entsprechender Ausgangsknoten verwendet. Von diesen aus werden nun (über *normalization*-Kanten) Types aufgesucht. Anschließend wird von dort aus zu den einzelnen Token traversiert (über *type_token*-Kanten), um danach noch zu deren Nachbartoken zu springen (über *next*-Kanten in beide Richtungen). Die für all diese Traversierungsoperationen angesprungenen Elemente des Graphen werden dann als Subgraph interpretiert, welcher in serialisierter Form versendet wird.

```

1 # -*- encoding : utf-8 -*-
2 words = params[:words].split(",")
3 type_nodes = @corpus.v_lookup(:string_value, words).in_e(:normalization).out_v
4 token_nodes = type_nodes.out_e(:type_token)
5 neighbor_traversal = token_nodes.in_v.branch{|r|r.in_e(:next).out_v}.branch{|r|
6   r.out_e(:next).in_v}.merge
7 out.data neighbor_traversal.subgraph.export

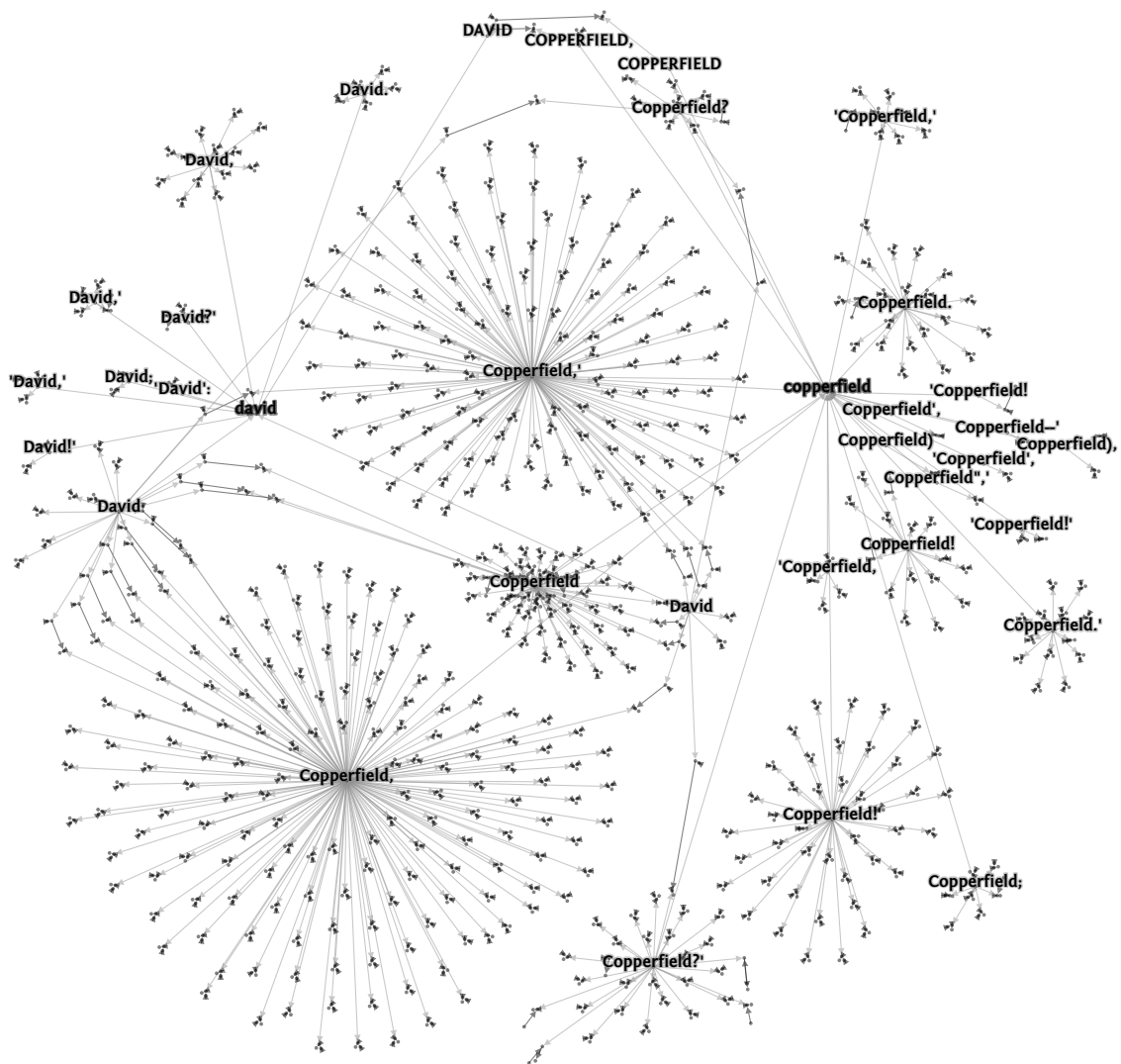
```

Quelltext 3.4: Anlegen einer neuen API-Definition

Die Nutzung dieser Subgraph-Funktionalität ist auch der Grund, warum die Traversierungsvorschriften so detailliert beschrieben werden müssen und z. B. explizit Kanten über *in_e* und *out_e* angesteuert werden. Die Route in Zeile 5 könnte in normalen Abfragen auch einfach als `token_nodes.both(:next)` geschrieben werden. Im Beispiel oben würden dann allerdings die Kanten nicht mitexportiert werden (was durchaus in einigen Anwendungsfällen gewünschte Funktionalität sein kann).

Abbildung 3.17 auf der nächsten Seite zeigt eine Visualisierung des Resultats eines API-Aufrufs mit den übergebenen Wörtern *david* und *copperfield* auf der Grundlage des *COPPERFIELD-Korpus*. Es zeigt sich bereits durch oberflächliche Größenvergleiche in der Grafik, wie sich die quantitative Verteilung der nicht normalisierten Types im Text ergibt. Die Zeichenkette *Copperfield*, ' kommt deutlich häufiger als Token im Text vor, als das reine *Copperfield*, was auf eine erzählende Schreibweise mit vielen eingeschobenen wörtlichen Reden hindeutet. Der Vorname kommt deutlich seltener vor als der Nachname und auch die Kombination beider Terme hintereinander (sichtbar durch entsprechende Verknüpfungen im Graphen) ist selten. Das spricht ebenfalls für die hauptsächliche Wiedergabe von wörtlicher Rede (da dort häufiger Anreden, wie „Mr. Copperfield“ vorkommen). Für detailliertere Interpretationsansätze müssten die ein-

⁴³Diese Anfrageart ist nicht standardmäßig in Pacer enthalten, wurde aber innerhalb von Kadmos hinzugefügt.



3.10.2 Gekapselte Erweiterung mit dem Plugin-System

Plugins sind nachladbare Programmteile, die zur Ausführung auf das Stammprogramm angewiesen sind und die dessen Funktionalität nutzen und erweitern können. Das hier vorgestellte Plugin-System ist dabei stark von der flexiblen Plugin-Architektur in Elasticsearch inspiriert.

In Kadmos erhalten Plugins Vollzugriff auf alle Objekte und Funktionen der Software - sie können also auch beliebige Daten der Datenbank lesen und schreiben. Damit ein breites Spektrum an Funktionen über Plugins realisiert werden kann, müssen sie zudem Vollzugriff auf die Laufzeitumgebung (und damit auf Funktionen des Betriebssystems) erhalten. In Linux-Umgebungen sollte dabei unbedingt eine Begrenzung des dadurch ermöglichten Schadenpotentials über Nutzer(gruppen)-Zugriffsrechte erfolgen. Für Installationen mit kleinem Datenumfang bietet sich grundsätzlich die Ausführung in einem virtualisierten System (isolierte Ausführung als „Gast“) an. In jedem Fall ist bei der Einbindung von Plugins Vorsicht geboten. Sie sollten (wie im Grunde jede Software) nur von vertrauenswürdigen Quellen bezogen und vor der Installation auf Integrität geprüft werden. Für die Installation von Plugins besteht noch kein automatisiertes System. In ein solches könnten Mechanismen zur elektronischen Signatur von Plugin-Versionen eingebaut werden. Einen letztendlichen Schutz vor Kompromittierung (oder ganz allgemein Fehlfunktionen) kann aber auch ein solcher Mechanismus nicht bieten.

Auch wenn bei der Nutzung von Kadmos-Plugins Sicherheitsbedenken hinsichtlich der Ausführung beliebigen Ruby- bzw. Java-Codes berücksichtigt werden müssen, so stellt diese Offenheit jedoch die einfachste und flexibelste Variante dar, um eine Plugin-Entwicklung und -Nutzung mit kurzen Release-Zyklen zu ermöglichen. Die Auslieferung erfolgt grundsätzlich im Quelltext, so dass die genaue Funktionsweise des Plugins daher jederzeit eingesehen werden kann. Allerdings besteht auch die Möglichkeit zum Hinzufügen ausführbarer Dateien, Java-Archiven oder anderer Programmbibliotheken. Im Zweifelsfall sollte also auf die Einbindung externer Plugins, Programme und Programmteile verzichtet werden – so, wie auch bei Webbrowser-Plugins, Smartphone-Apps usw. vorgegangen werden sollte, wenn sie Berechtigungen verlangen, die man ihnen nicht zugestehen möchte.

Alle Kadmos-Plugins werden in einem jeweils eigenen Unterverzeichnis des Ordners `./plugins/` abgelegt. Dadurch ist es möglich, ihre Entwicklung direkt im laufenden System unter Nutzung eines eigenen isolierten Quelltext-Repositorys bzw. Versions-

kontrollsystems durchzuführen. Plugins können – durch eine Art *Namespace*-Konzept von den regulären Endpunkten abgekapselt – eigene API-Endpunkte erstellen. Daneben können HTML-Oberflächen für die interaktive Ansteuerung des Plugins im Browser gestaltet werden.

Im Folgenden wird kurz dargelegt, wie eine externe Softwarekomponente in Form eines Plugins prinzipiell in Kadmos integriert werden kann. Die gewählte Beispiel-Komponente ist der Mate Tagger,⁴⁴ wie in [BN12] beschrieben. Es handelt sich dabei um einen POS-Tagger (und Abhängigkeitsparser), der durch hinzuladbare „Modelldateien“ für die Verarbeitung verschiedener Sprachen eingesetzt werden kann. Es soll damit auch gezeigt werden, dass Ergebnisse von Text- und Sprachanalysen nicht nur einmalig beim Einlesen der Texte physisch in der Datenbankstruktur persistiert, sondern auch *on-the-fly* zur Abfragezeit berechnet werden können. Da zur Ermittlung des POS-Tags eines einzelnen Worts üblicherweise längere Sequenzen benötigt werden – meist der ganze Satz – ergeben sich zwar entsprechende Mehraufwände für die Abfrage, auf eine Vorberechnung des Tags kann jedoch dank der flexiblen Datenhaltung auch hier verzichtet werden.

Im Sinne einer schnellen prototypischen Integration ist es meist nicht sinnvoll, eventuell anfallende Anpassungen der Komponente in ihrem Quelltext vorzunehmen. Dies würde bei jeder Änderung zu neuen Kompilierungs- und Integrationsaufwänden (incl. Neustart des Kadmos-Systems) führen. Eine leichtgewichtige Form der Integration von Drittanbieter-Software lässt sich z. B. über das Ausführen von Kommandozeilenbefehlen in JRuby als Subprozess realisieren. Die Kommunikation mit diesen lässt sich über den programmatischen Zugriff auf die Standard-Datenströme `stdin`, `stdout` und `stderr`, über Sockets oder über (temporäre) Dateien bewerkstelligen. Nachteilig sind dabei die ggf. großen Aufwände für das Erzeugen und Initialisieren des neuen Prozesses.

Der Mate Tagger ist ein Java-Programm, für dessen Start eine komplette Java-Laufzeitumgebung initialisiert werden muss. Entsprechende Verzögerungen im Sekundenbereich sind beim oft wiederholten wahlfreien Taggen kurzer Sequenzen nicht hinnehmbar. Durch die Verwendung von JRuby ist es jedoch möglich, die binäre Distribution des Taggers als *Java Archive* direkt einzubinden und in der selben Laufzeitumgebung, in der auch Kadmos ausgeführt wird, laufen zu lassen. Der direkte Zugriff auf die Programmfunktionen ermöglicht dann eine erweiterte Nutzung, etwa eine parallele Ausführung mehrerer Tagging-Vorgänge. Anpassungen lassen sich leicht durch Vererbungsmechanismen und alternative Methodenaufrufe der beteiligten Java-Klassen umsetzen, soweit das im Ori-

⁴⁴<http://code.google.com/p/mate-tools/>

ginalquelltext vorgesehen ist. Im Fall des Mate Taggers muss die Haupt-Funktion des Tagging-Aufrufs durch den sequenziellen Aufruf einzelner Teilschritte nachimplementiert werden, was allerdings nur wenige Zeilen Programmcode erfordert. Der Grund für den Anpassungsaufwand ist, dass die Methoden auf ein dateibasiertes Laden von Daten ausgerichtet sind, was für die Anwendung aus Performancegründen durch das Laden aus einer Stringrepräsentation im Hauptspeicher ersetzt werden sollte.

Der folgende kurze Quelltextausschnitt demonstriert, wie über Mechanismen der in JRuby leicht zugänglichen *Java-Reflection* dabei auch auf solche Programmelemente (z. B. Felder und Methoden) zugegriffen werden kann, die eigentlich nicht für eine Nachnutzung vorgesehen sind:

```

50 static_cend = ExtractorT2.java_class.declared_fields.find{|f|f.name=="_CEND"}
51 static_cend.accessible = true
52 instances_tagger.fill_chars instance, 0, static_cend.static_value.to_s.to_i
53 tag = java_class.declared_method :tag, InstancesTagger, SentenceData09
54 tag.accessible = true
55 tag.invoke self, instances_tagger, instance

```

Quelltext 3.5: Ausschnitt aus der Tagger-Wrapperklasse

Hier wird der Zugriff auf zwei zur Ausführung benötigte Elemente bestehender Klassen durch das Setzen der Eigenschaft „accessible = true“ auf den per *Reflection* erlangten Repräsentanten ermöglicht. Dieser Umweg ist erforderlich, da die Elemente in ihren Klassen im Java-Quelltext durch die Ausdrücke „static int _CEND“ und „private void tag“ deklariert wurden (und damit nicht „public“ sind). Die in Kadmos verwendete Technologie erlaubt auf diese Weise die flexible Nutzung der Funktionen von kompilierten Java-Programmen, selbst in Fällen, in denen eine Nutzung als Programmbibliothek nie vorgesehen war.⁴⁵

Der so integrierte Mate Tagger lässt sich nun nach Belieben mit Textabschnitten aus der Datenbank speisen. Er liest dabei Sätze tokenweise ein und gibt anschließend für alle Positionen eine Reihe von Analyseergebnissen aus. Hier wird der Begriff „Token“ im Sinne der Segmentierung des POS-Taggers genutzt. Das bedeutet, dass Interpunktionszeichen (und Sequenzen davon) für ein korrektes Tagging als ein eigenständiges Token behandelt werden müssen. Durch die Speicherung aller Sonderzeichen in korrekter Sequenz innerhalb unnormalisierter Types kann in Kadmos der dafür nötige Strom von „Tagger-Tokens“ effizient erzeugt werden.

⁴⁵es sei denn, es wurden aktiv Maßnahmen ergriffen, um dies zu verhindern, etwa durch *Code Obfuscation*

Die Analyseergebnisse beinhalten Informationen zu Wortart, Numerus, Genus, Tempus und weiteren grammatikalischen Kategorien. Die Ergebnisse werden in eine **JSON**-Repräsentation übertragen, wobei nur werttragende Eigenschaften berücksichtigt werden, so dass die Beschreibung kompakt ist. Für ein Satzzeichen ergibt sich so z. B. der Wert `{pos:"u"}`, da eine weitere Unterscheidung, etwa in Singular oder Plural bei Interpunktion nicht sinnvoll ist. Die Sammlung von so analysierten Token wird an den Client gesendet, wo eine Visualisierung stattfinden kann.

Das im Folgenden genutzte Modell für das **POS**-Tagging von altgriechischem Text wurde von Celano und Kollegen entwickelt, s. [CCM16], und dem Autor freundlicherweise bereits vorab für diese Arbeit zur Verfügung gestellt. In Abbildung 3.18 wird für die Suche nach „Κάδμου“ im **TLG-Korpus** gezeigt, welche Rückgabe den Browser erreicht und welche visuelle Aufbereitung mit entsprechend formatierten **HTML**-Elementen darauf aufbauend möglich ist. Die Beispielstelle entstammt dem Buch 4 von Diodors „*Bibliotheca Historica*“⁴⁶.

```
{
  "intermediate_result": {
    "doc": "tlg:0060:001 | Book 4, chapter 66t, section 3t, line 5t - Book 4, chapter 66t, section 3t, line 11t",
    "tokens": [
      {
        "token": "Ἀφροδίτης",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "n"
        }
      },
      {
        "token": "γάρ",
        "grammar": {
          "pos": "d",
          "token": ",",
          "grammar": {
            "pos": "u"
          }
        }
      },
      {
        "token": "ὥς",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "φασί",
        "grammar": {
          "pos": "v",
          "per": "3",
          "num": "p",
          "tem": "p",
          "mod": "i",
          "voc": "a"
        }
      },
      {
        "token": "τὸ",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "παλαιὸν",
        "grammar": {
          "pos": "a",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "δωρησαμένης",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "f",
          "cas": "g"
        }
      },
      {
        "token": "Ἀρμονίᾳ",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "f",
          "cas": "d"
        }
      },
      {
        "token": "τῇ",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "f",
          "cas": "d"
        }
      },
      {
        "token": "Κάδμου",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "g"
        }
      },
      {
        "token": "τόν",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "g"
        }
      },
      {
        "token": "τε",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "ὄρμον",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "καὶ",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "πέπλον",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "ἄμφότερα",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "ταῦτα",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "προσδέξασθαι",
        "grammar": {
          "pos": "v",
          "per": "3",
          "num": "p",
          "tem": "p",
          "mod": "i",
          "voc": "a"
        }
      },
      {
        "token": "τὴν",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "f",
          "cas": "a"
        }
      },
      {
        "token": "Ἐριφύλην",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "f",
          "cas": "a"
        }
      },
      {
        "token": "τὸν",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "μὲν",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "ὄρμον",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "παρὰ",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "Πολυνείκους",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "λαβοῦσαν",
        "grammar": {
          "pos": "v",
          "per": "3",
          "num": "p",
          "tem": "p",
          "mod": "i",
          "voc": "a"
        }
      },
      {
        "token": "τὸν",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "δὲ",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "πέπλον",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "n",
          "cas": "a"
        }
      },
      {
        "token": "παρὰ",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "τοῦ",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "υἱοῦ",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "τοῦ",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "Πολυνείκους",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "Θερσάνδρου",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "ὅπως",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "πείσῃ",
        "grammar": {
          "pos": "v",
          "per": "3",
          "num": "p",
          "tem": "p",
          "mod": "i",
          "voc": "a"
        }
      },
      {
        "token": "τὸν",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "υἱὸν",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "m",
          "cas": "a"
        }
      },
      {
        "token": "στρατεύειν",
        "grammar": {
          "pos": "v",
          "per": "3",
          "num": "p",
          "tem": "p",
          "mod": "i",
          "voc": "a"
        }
      },
      {
        "token": "ἐπὶ",
        "grammar": {
          "pos": "c"
        }
      },
      {
        "token": "τὰς",
        "grammar": {
          "pos": "1",
          "num": "s",
          "gen": "f",
          "cas": "a"
        }
      },
      {
        "token": "Θήβας",
        "grammar": {
          "pos": "n",
          "num": "s",
          "gen": "f",
          "cas": "a"
        }
      }
    ]
  }
}
```

Ἀφροδίτης γάρ , ὥς φασί , τὸ παλαιὸν δωρησαμένης Ἀρμονίᾳ τῇ Κάδμου τόν τε ὄρμον καὶ πέπλον , ἀμφότερα ταῦτα προσδέξασθαι τὴν Ἐριφύλην , τὸν μὲν ὄρμον παρὰ Πολυνείκους λαβοῦσαν , τὸν δὲ πέπλον παρὰ τοῦ υἱοῦ τοῦ Πολυνείκους Θερσάνδρου , ὅπως πείσῃ τὸν υἱὸν στρατεύειν ἐπὶ τὰς Θήβας :

Abbildung 3.18: Übertragung intermediärer **POS**-Annotationen im **JSON**-Format (oben, Auszug) und Ausgabe nach Übertragung der Grammatikinformationen zu **CSS**-Regeln (unten)

Der Mate Tagger erzeugt (ein entsprechend trainiertes Modell vorausgesetzt) auch ein Netzwerk von grammatikalischen und inhaltlichen Abhängigkeiten, einen so genannten **Dependenzgraphen**. Für dessen Visualisierung kann z. B. die entsprechende Client-Komponente der Annotationssoftware **brat**⁴⁷ oder eine auf Baumstrukturen ausgelegte

⁴⁶Die Belegstelle kann außerhalb des **TLG-Korpus** in freier digitaler Edition z. B. hier gefunden werden:
<http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:2008.01.0540:4:66:3>

⁴⁷<http://brat.nlplab.org/>

Visualisierungsmethode nachgenutzt werden. Daneben bietet sich ggf. auch eine generische – z. B. kräftebasierte – Visualisierung des Graphen an.

Mit diesem Beispiel wurde gezeigt, wie die Einbindung von Fremdkomponenten zur Textanalyse in Kadmos problemlos und effizient möglich ist. Daneben existieren natürlich viele weitere Möglichkeiten zur Erweiterung des Systems. Ein Plugin zum direkten Bearbeiten von [API](#)-Endpunkten im Browser über die Client-Bibliothek CodeMirror⁴⁸ ist derzeit in Entwicklung – doch es können nicht nur neue Oberflächen und Funktionen für menschliche Nutzer erstellt, sondern auch alternative Webservices über Plugins realisiert werden. Die Möglichkeit zur Nutzung beliebigen Ruby-Codes erlaubt es z. B., die [XML](#)-basierte Schnittstellenbeschreibung und -kommunikation entsprechend einer [SOAP](#)-basierten Webservicearchitektur mit [WSDL](#)-Schicht mit vergleichsweise geringem Aufwand umzusetzen. Es wurden zudem bereits erste Experimente zur Implementierung einer [OAI-PMH](#)-Schnittstelle für bessere Auffindbarkeit der gespeicherten Korpora (bei vollständiger Interoperabilität mit bestehenden Systemen) durchgeführt.

Über das Plugin-System ist es darüber hinaus prinzipiell auch möglich, eine native graphische Benutzeroberfläche zu erzeugen. Dazu können über JRuby beliebige „*Window Toolkits*“, beispielsweise das in Java bereits enthaltene *Swing*, angesteuert werden. Damit bietet sich über die Plugin-Schnittstelle eine Alternative zum webbrowsergestützten Zugriff auf die Funktionalität von Kadmos. Eine solche Erweiterung erlaubt zum Beispiel die Verwendung von Visualisierungs-Bibliotheken, die ausschließlich für den Einsatz in Java-Desktop-Umgebungen geschrieben wurden.

Nachdem nun die verschiedenen Voraussetzungen für eine flexible Erweiterung vorgestellt wurden, kann nachfolgend untersucht werden, wie sich die entwickelte Technologie zur Bearbeitung breiterer Fragestellungen in den e-Humanities nutzen lässt. Im nächsten Kapitel werden dafür zunächst allgemeine Überlegungen zur Nutzung von Kernkomponenten zur Erstellung neuer Systeme vorgenommen, um anschließend praktische Beispiele vorzustellen.

⁴⁸<http://codemirror.net/>

Kapitel 4

Modellerweiterungen und komplexere Anwendungsfälle

*Models containing more information
are always more powerful
than those containing less [...]*

Wayne W. Zachary

US-Amerikanischer Anthropologe und Informatiker

aus dem Artikel

„An Information Flow Model for Conflict and Fission in Small Groups“, [Zac77]

4.1 Erweiterbarkeit und Konstruktive Voraussicht

Die zentrale Grundanforderung an die in dieser Arbeit entwickelte Technologie wurde bereits in Abschnitt 2.5 auf Seite 69 herausgearbeitet: Das erstellte System muss genügend Flexibilität für eine Nutzung in erweiterten, angrenzenden oder auch komplett neuen Verwendungskontexten aufweisen. Bis hierhin wurden für den Umgang mit Textdaten bereits drei Merkmale des Systems vorgestellt, die auf diese Flexibilitätsanforderung ausgerichtet sind:

- Ein offenes, jederzeit änderbares Datenmodell, das auf einer stetig populärer werdenden Technologie aufsetzt,
- Basiskomponenten für den Betrieb einer plattformunabhängig lauffähigen, robusten und skalierbaren Anwendung für verteilte Zugriffe auf Forschungskorpora und
- eine Vielzahl von Entwicklungsmöglichkeiten für extrem kurze Iterationen bei der funktionalen Erweiterung über Echtzeit-Änderungen an den bereitgestellten API-Funktionen bei voller Unterstützung des kompletten Ökosystems von Java-Bibliotheken und (JRuby-kompatiblen) *Gems*.

Es wurde gezeigt, dass damit übliche Aufgabenstellungen im Umfeld des Text Mining und der explorativen Recherche von Textdaten aus einem neuen Blickwinkel betrachtet und innerhalb geänderter Rahmenbedingungen mit neuen oder angepassten Verfahren bearbeitet werden können. In diesem Kapitel wird die Nutzbarkeit des Systems in erweiterten Anwendungsszenarien auf den Prüfstand gestellt.

Bevor die einzelnen Anwendungsfälle beschrieben werden, soll hier noch kurz der Nutzen eines weitgehend generischen, auf Erweiterbarkeit ausgelegten (und damit für die konkrete erweiterte Anwendung potentiell „unfertigen“) Systems motiviert werden: Warum lohnt es sich, angesichts in jedem Fall nötiger Entwicklungsaufwände in konkreten Anwendungsprojekten, auf ein solches Basissystem zurückzugreifen?

Für die Beantwortung dieser Frage bietet sich eine Betrachtung aus ingenieurtechnischer Sicht an. In der Konstruktionslehre ist die Unterscheidung von Entwicklungsvorgängen nach so genannten Konstruktionsarten (*types of design*) ein bereits lange etabliertes Konzept, vgl. [PB86].

Es wird dort unterschieden in:

- Neukonstruktion, die Erarbeitung eines neuen Lösungsprinzips für bekannte oder neue Problemstellungen,
- Anpassungskonstruktion, das Weiterverwenden eines bestehenden Systems unter Abänderung vorhandener Komponenten und eventueller Neukonstruktion von Teilkomponenten und
- Variantenkonstruktion, das Variieren genau spezifizierbarer Parameter eines „vorgedachten“ Systems unter Beibehaltung von Funktion und Lösungsprinzip

Es ist leicht ersichtlich, dass Neukonstruktionen so oft es geht vermieden werden sollten. Sie bringen die größten Aufwände und Projektrisiken mit sich und profitieren zudem nicht von den in bewährten Lösungsprinzipien enthaltenen vorausgehenden Erfahrungen und *best practices*.

Die Variantenkonstruktionen erscheinen aus dieser Sichtweise am erstrebenswertesten, da sie den geringsten Aufwand erfordern und die schnellste Einsatzmöglichkeit für ein neues so erstelltes Werkzeug versprechen. Jedoch hat die Betrachtung des Forschungskontexts der e-Humanities gezeigt, dass die Erstellung allumfassender generischer Werkzeuge weder möglich, noch (im Sinne eines unvoreingenommenen, offenen Forschungsprozesses) in jedem Falle gewünscht ist.

Die oben beschriebenen Flexibilitätsmerkmale von Kadmos positionieren das System in das Kontinuum zwischen Varianten- und Anpassungskonstruktion: Für einfache, oft wiederkehrende Arbeitsschritte ist es problemlos möglich, das Basissystem in einer geeigneten Konfiguration (Variante) ohne zusätzliche Entwicklungsaufwände mit neuen Quellensammlungen und Nutzeroberflächen zu betreiben und dabei die zur Verfügung gestellte Basisfunktionalität zu nutzen. Für Aufgaben, die sich von Beginn an komplexer darstellen, oder für die bei der Variation des Basissystems erkannt wird, dass zusätzliche Anforderungen bestehen, hält Kadmos alle benötigten Erweiterungsmöglichkeiten bereit. Die bereits beschriebene Schaffung der entsprechenden technischen Möglichkeiten entspricht der – unter der Bezeichnung „Konstruktive Voraussicht“ geführten –perspektivisch geplanten Erweiterbarkeit von Systemen.

Das gewünschte Vorgehensmodell einer inkrementellen prototypischen Entwicklung kann damit von Projektbeginn an umgesetzt werden. Es kommt zu keinem „Leerlauf“

auf fachwissenschaftlicher Seite während der Entwicklung einer ersten lauffähigen Version des Recherchewerkzeugs und gleichzeitig zu keinem „Vorlauf“ der Technologieentwicklung, während welchem sich gegebenenfalls vorhandene Missverständnisse von Entwicklungsziel und Analyseaufgaben längerfristig im Kern des erstellten Werkzeug manifestieren können.

Der Vorteil eines einfachen aber flexiblen Basissystems ist, dass es Einschränkung durch zu scharf abgegrenzte Vorgaben hinsichtlich Schema und Abfragemodi mit sich bringt. Gleichzeitig hilft das einfache Datenmodell, einen zu hohen initialen Abstraktionsgrad zu vermeiden, welcher einen zu starken Formalisierungsgrad der abzubildenden Arbeiten voraussetzen würde. Vor solchen Tendenzen wurde bereits in [SM99] gewarnt:

[...] creators of systems that support intellectual work like design, writing, or organizing and interpreting information are particularly at risk of expecting too great a level of formality from their users.

Das richtige Maß sollte sich folglich während der Projektarbeit durch sukzessive Steigerung des Formalisierungsgrades (und damit Erweiterung und Spezialisierung des Datenmodells und der Programmfunktionalität) ergeben – angefangen bei der Variierung des Systems hin zur Erweiterung, sofern sie tatsächlich benötigt wird.

Weil die einzelnen Projektanforderungen grundsätzlich variabel sind, ist es in diesem Rahmen schwer, pauschal den Nachweis zu erbringen, dass die getroffene Technologiewahl, das Systemdesign und das avisierte prototypzentrische Vorgehensmodell zweckmäßig sind. Die exemplarische Demonstration einer Bandbreite von abdeckbaren Themen und Methoden ermöglicht es jedoch zu untersuchen, ob es bei der Bearbeitung konkreter Fragestellung hilfreich eingesetzt werden kann, und dabei den in Abschnitt 3.1 auf Seite 74 vorgestellten unterschiedlichen Prototypen-Begriffen gerecht wird.

In den folgenden Abschnitten werden Beispiele aus dem Forschungsalltag vorgestellt, für die Lösungen innerhalb des im vorangehenden Kapitel eingeführten Modellierungsparadigmas und unter Verwendung der bis hierhin vorgestellten Basistechnologien umgesetzt wurden. Mit den präsentierten Lösungen wird kein Anspruch auf eine optimale Behandlung der Problemstellung erhoben. Weder stand ein solcher Anspruch in der Konzeptionsphase im Vordergrund, noch waren entsprechende Evaluierungen und Optimierungen zeitlich und inhaltlich möglich. Wichtigstes Ziel dieses Kapitels ist die Demonstration der

Erweiterungsmöglichkeiten des Kadmos-Systems und der Flexibilität der zugrundeliegenden Technologien.

In der Tat zeigt sich aber, dass die Nutzung von Graphdatenbanken für den Großteil der betrachteten Problemstellungen kein „unnatürlich“ wirkender Kunstgriff, sondern eine durchaus passende Form der Abbildung und ein geeigneter Modus des Zugriffs für die Daten der jeweiligen Anwendungsdomäne ist. Die Analyse lokaler Kontexte ist ein zentraler Ankerpunkt vieler Untersuchungen in den Sozial- und Geisteswissenschaften. Die direkte Verfügbarkeit solcher Kontexte in der Graphenrepräsentation fördert einen informierten, agilen und zu den Daten kohärenten Forschungsprozess.

Im Folgenden wird u. a. gezeigt, wie damit die Vorbereitung von Netzwerkanalyse, die inhaltliche Alignierung von mehrsprachigen Texten, die Analyse verschiedener Ebenen von Sequenzen und Strukturierung, die Erstellung von Beschleunigungsstrukturen und darüber hinaus auch eine Unterstützung des Kommunikationsprozesses in interdisziplinären Projekten möglich ist.

4.2 Entitäten-Netzwerke

4.2.1 Navigationsunterstützung und Netzwerkanalyse

Der Inhalt von Texten erschließt sich meist nicht nur durch die darin beschriebenen abstrakten Konzepte, sondern insbesondere auch über die darin vorkommenden konkreten (realen wie fiktionalen) Dinge, Personen, Institutionen usw. – kurz: Entitäten. Besitzen diese Entitäten einen Namen und ist dieser auch außerhalb des Textes bekannt und für eine (mehr oder weniger) eindeutig identifizierbare Entität gebräuchlich, so kann dieser Name auf lexikalischer Ebene als Ankerpunkt für externes Wissen gelten, welches bei der Interpretation des Textes nützlich und teils sogar unverzichtbar ist.

Daher ist bei der Verarbeitung großer Textkollektionen die automatische Eigennamenerkennung ein zentrales Werkzeug zur Herstellung von Verknüpfungen zu Wissensbasen. Die unter der Bezeichnung **Named Entity Recognition (NER)** bekannten Verfahren stützen sich u. a. auf Wortlisten und Enzyklopädien, Regelbeschreibungen zu häufigen Mustern von Wörtern und bekannten Phrasen im Umfeld von Entitätennamen sowie auf linguistische Annotationen und Satzparsing.

Die meisten Verfahren sind dabei nicht nur in der Lage, benannte Identitäten im Text zu finden, sondern können zusätzlich auch den entsprechenden Entitätentyp angeben (*Named Entity Classification*). Die Granularität der Klassifizierung ist bei dieser Aufgabe oft nicht genau spezifiziert: Die Zeichenkette „Universität Leipzig“ kann z. B. als Organisation, Institution, Bildungseinrichtung oder Universität beschrieben werden. Hier zeigt sich eine große Nähe zur hierarchischen Modellierung der Klassen von Aussagegegenständen in Ontologien. Meist beschränken sich NER-Verfahren hierbei auf wenige und sehr allgemeine Basistypen, wodurch das Klassifizierungsproblem vereinfacht wird und wodurch gleichzeitig pro Klasse Beispiele für statistische Musteranalysen zur Verfügung stehen.

Benennung von Identitäten sollte hier nicht mit Identifizierung verwechselt werden, da die Namen selten eindeutig sind. Die Identität von Entitäten wird durch deren Benennung im Kommunikationskontext zwar angedeutet, ist aber dadurch nicht zwangsläufig auf eine universelle Wissensbasis abbildbar. Bei mündlicher Kommunikation wird die Identität der Gesprächsgegenstände ggf. durch Nachfragen „ausgehandelt“, wohingegen in der schriftlichen Kommunikation oft versucht wird, genügend Beschreibungen und Zusatzinformationen zur Erleichterung einer Identifizierung mit anzugeben. Dabei werden jedoch meist Vermutungen über das Wissen des Kommunikationspartners (bei Publikationen die der Mehrheit der Leserschaft) zugrunde gelegt, die bei der Analyse nur schwer greifbar sind. Daher existieren immer Mehrdeutigkeiten (sogenannte Ambiguitäten), für deren konkrete Identifizierung eine ausführliche Kontextanalyse der Benennungen notwendig ist – die sogenannte Disambiguierung. Ist die Identität hinreichend geklärt, besteht die Problematik, diese eindeutig kommunizieren zu können (also ohne Benennung und Kontexte dabei explizit mit angeben zu müssen). Für diese Aufgabe bietet sich z. B. die im Semantic Web zur Referenzierung von Ressourcen etablierte Nutzung von URIs an. Auch andere Nummerierungsschemata, wie sie z. B. Normdatenprovider verwenden, sind geeignet, um die Identität unter Angabe von Nummerierungssystem und Nummer zu gewährleisten.

An dieser Stelle kann nicht näher auf die Grundlagen oder einzelnen technischen Details von NER-Verfahren eingegangen werden. Stattdessen soll aber anhand konkreter Anwendungsfälle gezeigt werden, wie die Ergebnisse der Eigennamenextraktion im Kontext einer Graphenrepräsentation von Texten gewinnbringend genutzt werden können.

Zunächst sollen Eigennamen dabei als alternative Zugangsform zu Dokumentsammlungen betrachtet werden. Diese Sichtweise ist in Form von Registern durchaus bereits in

traditionellen Medien realisiert. Diese bilden eine Navigationshilfe für konkrete Recherchen zu Entitäten und schaffen eine einfache Form von Hypertextualität: Trifft der Leser in langen Dokumenten auf den Namen einer interessanten Entität, kann er über das Register zu weiteren Textstellen springen (meist sehr grobgranular über Seitenangaben), in welchen die Entität ebenfalls Erwähnung findet. Im Folgenden soll die semi-automatische Erstellung eines Personenregisters mit Hilfe von [NER](#)-Verfahren vorgestellt werden.

Im konkreten Fallbeispiel werden automatische Verfahren zur inhaltlichen Erschließung des [REKTOREN-Korpus](#) eingesetzt. Dieses Korpus wurde vom Leipziger Universitätsarchiv zur Verfügung gestellt. Es enthält 125 Reden Leipziger Universitätsrektoren. Lange Zeit war es üblich, zum jährlich stattfindenden Rektoratswechsel eine feierliche Zeremonie abzuhalten, im Rahmen derer der scheidende Rektor eine Rede in Form eines Jahresberichts hält und der neugewählte Rektor in seiner Antrittsrede allgemeine wissenschaftliche Impulse gibt – meist im Bezug auf sein eigenes Fachgebiet. Die Rektoratsreden wurden ab 1871 in gedruckter Form veröffentlicht.

Anlässlich des 600-jährigen Universitätsjubiläums wurden die Rektoratsreden von 1871 bis 1933¹ in einem gemeinsamen Band neu veröffentlicht, s. [[Häu09](#)]. Der Veröffentlichung ist eine Digitalisierung der alten Druckfassungen vorausgegangen. Insgesamt wurden 2340 Seiten gescannt und händisch oder mittels [OCR](#) in maschinenlesbaren Text umgewandelt². In manueller Nacharbeit wurden in den Texten Druckfehler, Übertragungsfehler und Fehler des [OCR](#)-Verfahrens behoben. Sonst wurde jedoch die damalige Rechtschreibung (und Grammatik) des Originals übernommen. Weitere Details sind in [[EBH15](#)] beschrieben.

Für die Umsetzung der Eigennamenerkennung existieren zahlreiche [NLP](#)-Frameworks. Die Entscheidung für die [General Architecture for Text Engineering \(GATE\)](#) ist hier praktischen Erwägungen geschuldet. In [[BS12](#)] wird die [Unstructured Information Management Architecture \(UIMA\)](#) als vielversprechende und flexible Architektur beschrieben, jedoch auch angemerkt, dass die Vielzahl nachnutzbarer Ressourcen ein Vorteil von [GATE](#) sind. Die im Gegenzug als Stärke von [UIMA](#) hervorgehobene Workflow-Steuerung in [UIMA](#), welche die Beschreibungssprache [BPEL](#) nutzt, wird im Kontext der Arbeiten am [REKTOREN-Korpus](#) hingegen nicht benötigt. Die Einbindung der [GATE](#)-Funktionalität und damit die Steuerung der Abläufe erfolgt direkt via JRuby.

¹Die nach 1933 zum Rektoratswechsel gehaltenen Reden wurden nicht mehr von gewählten, sondern von eingesetzten Rektoren gehalten und sind nicht für den Abdruck in unkommentierter Edition geeignet.

²Für in Fraktur gesetzte Dokumente waren zu dieser Zeit noch keine geeigneten [OCR](#)-Verfahren verfügbar.

Das Komponentensystem von GATE namens „a Collection of REusable Objects for Language Engineering (CREOLE)“ stellt mit A Nearly-New Information Extraction System (ANNIE) ein etabliertes Standardverfahren für die NER mit Wortlisten und Regeln dar. Alternativ oder zusätzlich dazu sind auch graphbasierte Verfahren denkbar (und in die vorgestellte Technologie grundsätzlich gut integrierbar), beispielsweise der „Pendel-Algorithmus“ aus [Bie03]. Im Fall der Rektoratsreden kann auf eine umfangreiche Ressource kontemporärer Vor- und Nachnamen aus dem digitalisierten Bestand des Universitätsarchivs zurückgegriffen werden. Zudem wurde eine Liste von abgekürzten Vornamen, wie „Ludw.“, „Joh.“ oder „Wilh.“, angelegt, wie sie damals üblich waren.

In ANNIE werden musterbasierte Regeln in so genannte *Transducer* umgewandelt, welche in der Automatentheorie beschrieben werden, und eine effiziente Abarbeitung der Mustersuche ermöglichen. Für die Regeldefinition stellt GATE die Beschreibungssprache Java Annotation Patterns Engine (JAPE) zur Verfügung. In dieser wurden typische in den Texten erkennbare Muster definiert, die auf die Nennung von Personen hinweisen.

Nach der Extraktion waren noch minimale Nacharbeiten notwendig: Zum einen musste eine Zuordnung von Duplikaten stattfinden. Dies umfasst z. B. eine Behandlung des Genitiv-Falls, wie die Zusammenführung von „Martin Luther“ und „Martin Luthers“ auf die erstere Variante. Weil die Texte z. T. eine veraltete Orthographie und andere als die heute üblichen Satzkonstrukte verwenden, konnten keine Standardwerkzeuge zum POS-Tagging oder für andere Verfahren zur grammatikalischen Analyse verwendet werden, mit denen diese Fälle direkt im NER-Prozess behandelt werden können. Andere Vereinheitlichungen waren nur in einem semi-automatischen Verfahren realisierbar, etwa die Vereinigung der Schreibweisen „Albrecht Mendelsohn-Bartholdy“, „Albrecht Mendelssohn“ und „Albrecht Mendelssohn-Bartholdy“. Zudem mussten einige fälschlicherweise ausgewählte Kandidaten-Zeichenketten manuell aussortiert werden, wie „Ernst des Lebens“, welche dem selben Muster wie die gewünschte Zeichenkette „Theodor des Coudres“ folgt. Für andere, wie „Ann Arbor“ fehlt dem Verfahren das nötige Hintergrundwissen – schon in [MMG99] wurde gezeigt, dass zur Erkennung benannter Entitäten geographischen Typs der textuelle Kontext meist nicht ausreicht.

Auf Identifizierung der Entitäten, z. B. die Zuweisung zu entsprechenden Wikipedia-Artikeln, wurde in diesem Anwendungsszenario verzichtet, da sie für viele der erkannten Personen trivial über den Eintragstitel möglich ist, für die anderen jedoch wesentlichen Rechercheaufwand bedeutet.

Für diese Form der Visualisierung erfolgte ein Export der Analyseergebnisse im Graph-Austauschformat GeXF³, wodurch die Nutzung von Programmen zur Graphanalyse (allen voran Gephi) möglich wird. In diesem Umfeld können auch verschiedene Methoden der Netzwerkanalyse angewendet werden. Es sind verschiedene Formen der Graphinduktion für den exportierten Graphen denkbar. Abbildung 4.2 zeigt den bereits vorgestellten Dokument-Entity-Graphen noch einmal im „freien“ Layout ohne Fixierung der Dokumentknoten. Das Einführen eines Kantengewichts-Schwellwerts passt die Visualisierung so an, dass nur Personen gezeigt werden, die mit mehr als einem Dokument verknüpft sind und die sich damit als Korpus-Navigations-Facette eignen.

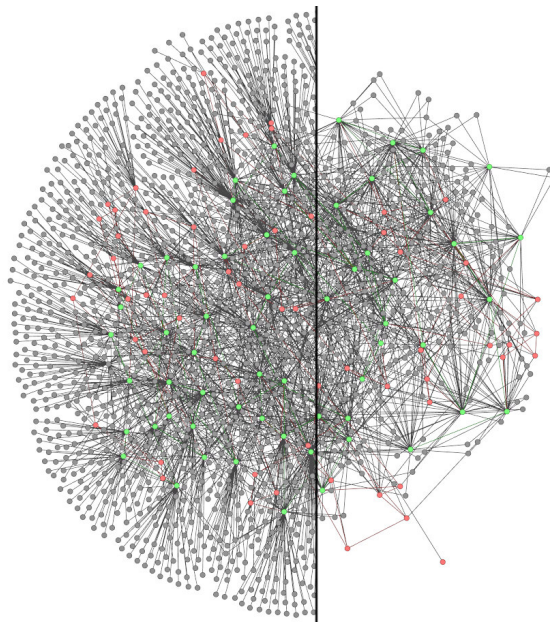


Abbildung 4.2: Dokument-Personen-Netzwerk, rechts ohne Personen mit Knotengrad 1

Neben diesem Datensatz kann auch die Dokumenten-Kookkurrenz von Entitäten zur Ausgabe eines Kookkurrenzgraphen ausgewertet werden. Das ermöglicht perspektivisch die Analyse thematischer oder sozialer Nähe, indem eine Person-Person-Kante als „virtuelle“ Kante aus dem Textmodell abgeleitet wird. Dabei muss die Interpretation des Ergebnisses behutsam erfolgen. Es ist beispielsweise nicht gerechtfertigt, in diesem Zusammenhang von einem „sozialen Netzwerk“ zu sprechen, wie für ähnliche Analysen z. B. in [Gra14] angemerkt wird. Insbesondere wegen der großen Kookkurrenzeinheit ist im Einzelfall nicht unbedingt von semantischen Gründen für das gemeinsame Auftreten von Entitäten auszugehen. Für eine statistische Mittlung dieser möglichen Zufallseffekte sind 125

³<http://gephi.org/gexf/format/>

Dokumente dabei bei weitem noch nicht ausreichend.

Weitergehende Arbeiten zur Graphinduktion aus [NER](#)-Ergebnissen und die Nutzung des Graphen zur Korpusnavigation wurden im in [\[KEAH14\]](#) beschriebenen Kooperationsprojekt (in der Hauptsache durch Christoph Kuras) für das [STASI-Korpus](#) durchgeführt.

Abschließend muss – wieder im Sinne von [\[Gra14\]](#) – darauf hingewiesen werden, dass Netzwerkanalyse lediglich ein Werkzeug in einer Kette von Analysen und Perspektivwechseln sein kann und keineswegs ein solitäres Analyseverfahren, an das sich keine kritische Betrachtung mehr anschließen muss.⁴ Eine weiterer Anwendungsfall, bei dem Netzwerke benannter Entitäten zum besseren Gesamtverständnis eines Textkorpus beitragen können, wird im folgenden Abschnitt vorgestellt. Dabei werden statt Personennamen nun die im Text genannten Orte untersucht.

4.2.2 Erzeugung und Exploration von Toponymnetzwerken

Ortsbezeichner lassen sich oft eindeutig identifizieren und lokalisieren und erlauben damit eine logische Verbindung von Texteinheiten in ein üblicherweise kartesisches Konzept von Raum und mithin in ein kartographisch unterfütterbares Koordinatensystem. Netzwerke benannter Entitäten lassen sich darüber in einem geographischen Referenzrahmen „erden“ und in diesem erweiterten Kontext interpretieren.

Das folgende Fallbeispiel widmet sich den neunbändigen „Historien“ des Herodot von Halikarnassos, die im [HERODOT-Korpus](#) in griechischer Edition und englischer Übersetzung enthalten sind. Für den englischen Text existieren Annotationen benannter Entitäten, wobei geographische Entitäten mit einem *Gazetteer* verknüpft sind – einer Ortsnamensliste, in der meist neben Verweisen auf Namensvarianten von Orten auch die Koordinaten der konkreten Geo-Entitäten hinterlegt sind.

Für die antike Welt des Mittelmeerraumes existiert mit Pleiades⁵ ein sehr umfangreicher Gazetteer, auf welchen in den [TEI](#)-Auszeichnungen im [HERODOT-Korpus](#) verwiesen wird. Diese Verweise basieren auf den bereits in Abschnitt [2.3.5 auf Seite 61](#) kurz vorgestellten Ansätzen von [LOD](#). Über diese Identifier sind die Einträge auch für den Abgleich mit anderen, den [LOD](#)-Prinzipien folgenden und mit Pleiades verknüpften Gazetteers, wie

⁴„Le réseau, un moyen pas une fin“

⁵<http://pleiades.stoa.org/>

dem „iDAI.gazetteer“⁶ des Deutschen Archäologischen Instituts geeignet. Verschiedene Datenquellen mit geographischen Bezügen sind auf diese Weise miteinander verbunden. Mit dem Projekt Pelagios⁷ [SBdI14] wird versucht, Synergien in diesem Umfeld zu stärken, und neue Werkzeuge und Standards für den Umgang mit diesen Daten zu schaffen.

In diesem Rahmen ist zu erwähnen, dass derzeit auch erste Ansätze für die Nutzung von Property-Graph-Datenbanken zur Verwaltung und Abfrage von semantisch verknüpften Gazetteers entwickelt werden, wie beispielsweise eine studentische Seminararbeit von Manuel Bär an der Universität Zürich aufzeigt, s. [Bär16].

Für die geographische Analyse der Historien wurde eine kookkurrenzbasierte Netzwerk-induktion auf Grundlage des im TEI enthaltenen Markups durchgeführt. In den dort verzeichneten Abschnitten von jeweils ca. zwei Sätzen Länge wird das gemeinsame Auftreten von als Toponym annotierten benannten Entitäten ausgewertet. Die meisten dieser Entitäten verfügen über eine zugewiesene Pleiades-ID und damit über abrufbare Geo-Koordinaten.

Dabei ist anzumerken, dass ein automatisches Verfahren selbstverständlich nicht die detaillierte Erschließungstiefe erreichen kann, die nur aufwändig durch manuelle (oder stark nutzergesteuerte semi-automatische) Analysen und die genaue Annotation geographischer Zusammengehörigkeit herstellbar ist. Im Vorwort zu [BBPI16] beschreiben Barker, Bouzarovski und Isaksen den (auch auf das Beispiel des HERODOT-Korpus bezogenen) Arbeitsprozess für den Netzwerkansatz bei der Analyse geographischer Überlieferungen des Altertums als *„first carefully identifying the occurrence of places as and when they occur in the narrative, and then equally carefully assessing whether the place is mentioned in relations to another and, if so, what form that relationship takes.“*

An die Stelle sorgfältiger Abwägungen im Einzelfall tritt in dieser Arbeit eine pauschale, möglichst wenige Annahmen tätige Behandlung der Kookkurrenz von Toponymen. Alle interpretativen Schritte werden in die Phase nach der Erstellung des Netzwerks verschoben, wobei auch (und gerade) dann trotz distanzierter und aggregierter Sichtweise ein Rückbezug bis zur einzelnen Textstelle erfolgen muss.

Abbildung 4.3 auf der nächsten Seite zeigt die für diesen Zweck geschaffene Webanwendung, den „Ancient World Explorer“. Das Programm dient zur visuellen Darstellung des

⁶<http://gazetteer.dainst.org/>

⁷<http://pelagios.org/>

geographisch verankerten Toponymnetzwerks⁸ und ermöglicht eine Interaktion mit den Knoten und Kanten dieses Netzwerks. Es ist ferner damit möglich, Teilnetzwerke für beliebig große Abschnitte der Historien (von wenigen Abschnitten bis zu den vollen neun Büchern) zu erstellen. Als Hintergrundkarte wurde die Basiskarte des „Ancient World Mapping Center“⁹ verwendet, welche nur topographische und keine modernen Kartenelemente verwendet.

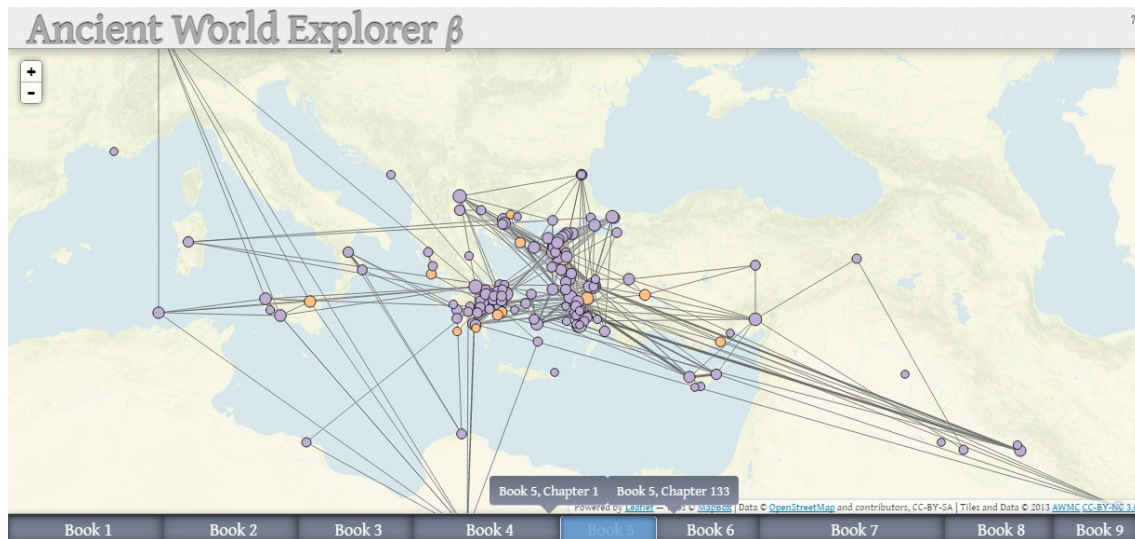


Abbildung 4.3: Ansicht des „Ancient Word Explorer“ mit geographisch verankertem Netzwerk von Toponymkookkurrenzen in Buch 5 des [HERODOT-Korpus](#)

Durch das Klicken auf einen Knoten wird ein Info-Kasten geöffnet, der den genaueren Entitätentyp (Insel, Siedlung, Hafen, Aquädukt), verschiedene Benennungen und einen Link zur Pleiades-Ressourcen-Seite enthält. Durch das Klicken auf eine Kante wird ein Kasten geöffnet, in welchem die englischen und griechischen Versionen derjenigen Textstellen aufgelistet sind, in welchen die durch die Kante verbundenen Toponyme gemeinsam auftreten. [Abbildung 4.4 auf der nächsten Seite](#) zeigt ein Beispiel dieser Ansicht samt der zur Textstelle zugehörigen [CTS-URN](#). Diese verweist wiederum per Hyperlink direkt zur entsprechenden Volltext-Ansicht im Perseus-Projekt¹⁰.

In [Abbildung 4.5 auf der nächsten Seite](#) ist überblicksartig die Sequenz der Toponymnetzwerke für alle neun Bücher der Historien abgetragen. Es zeigt sich, dass der lokale Fokus der Beschreibung über das gesamte Narrativ hin doch recht stark variiert – eine

⁸Zu diesem Zweck wurde existierende Basisfunktionalität der Javascript-Bibliothek D3 [\[BOH11\]](#) entsprechend angepasst.

⁹<http://awmc.unc.edu/>, weitere Information zur Karte hier:

<http://awmc.unc.edu/wordpress/tiles/map-tile-information>

¹⁰<http://www.perseus.tufts.edu/>

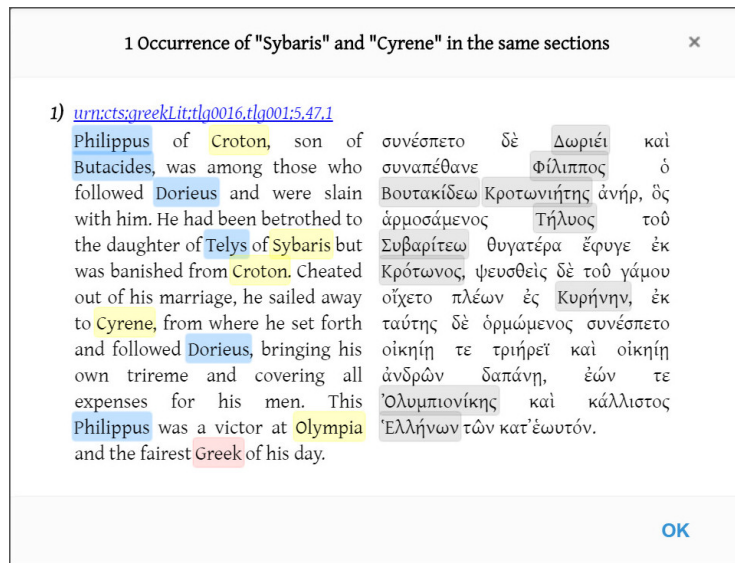


Abbildung 4.4: Detailansicht für Toponymkookkurrenz von Sybaris und Cyrene mit nach Typ eingefärbten Entitäten

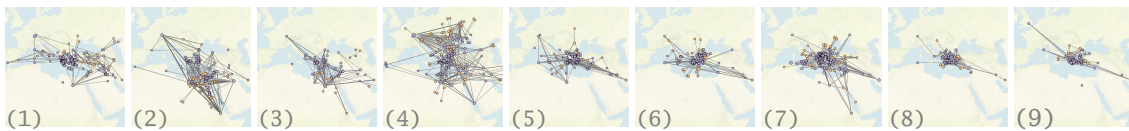


Abbildung 4.5: Sequenz von Toponymnetzwerken über alle Bücher im [HERODOT-Korpus](#)

Beobachtung, die natürlich in der Fachwissenschaft schon lange bekannt ist, die hier aber auch dem Laien eindrücklich dargelegt werden kann.

Die nicht mit Koordinaten ausgestatteten Knoten werden in der Ansicht durch einen kräftebasierten Graph-Layout-Algorithmus in der Nähe ihrer Kookkurrenten abgetragen. Die Grundannahme, dass eine Kookkurrenz von Toponymen nicht selten durch den aktuellen lokalen Fokus der Schilderung hervorgerufen wird, sorgt dafür, dass die anzunehmende reale Position in einigen Fällen nicht allzu weit von der so auf die Karte projizierten Kartenposition abweicht. Diese Sichtweise ist natürlich nicht immer zutreffend. Wird beispielsweise im Text eine Liste von griechischen Städten genannt, die unter dem gleichen Herrscher gegründet wurden, müssen diese nicht zwangsläufig nah beieinander liegen.

Abbildung 4.6 auf der nächsten Seite zeigt einen weiteren Fall, in dem die geographische Lage sich trotz dreier Kookkurrenten per „Triangulierung“ über die Kookkurrenzorte bestimmen lässt. Dieser Umstand fällt jedoch erst bei genauerer Ansicht der entsprechenden Textstelle (urn:cts:greekLit:tlg0016.tlg001:9.93.1) auf:

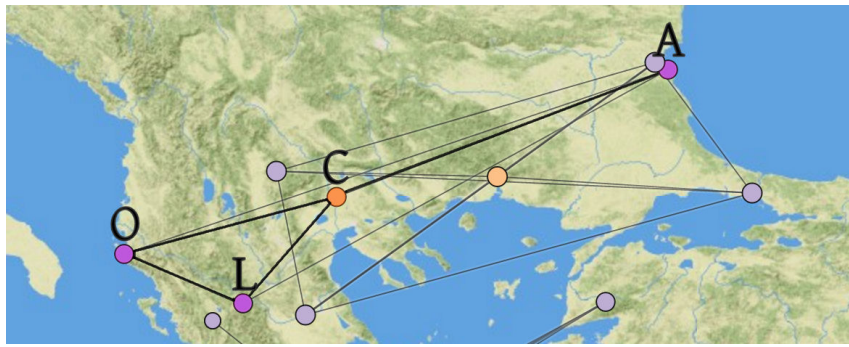


Abbildung 4.6: Netzwerkverbindungen und Kartenabbildung der annotierten Orte Apollonia, Chon, Lacmon und Oricum. Chon ist nicht georeferenziert.

*There is at **Apollonia** a certain flock sacred to the Sun, which in the daytime is pastured beside the river **Chon**, which flows from the mountain called **Lacmon** through the lands of Apollonia and empties into the sea by the harbor of **Oricum**. By night, those townsmen who are most notable for wealth or lineage are chosen to watch it, each man serving for a year, for the people of **Apollonia** set great store by this flock, being so taught by a certain oracle. It is kept in a cave far distant from the town.*

Während Apollonia, Lacmon und Oricum über den Gazetteer geographisch verknüpft sind,¹¹ ist für Chon nur bekannt, dass es sich dabei um einen Ortsbezeichner handelt – einen Fluss, wie man dem Text entnehmen kann. Nach einem Blick auf die Karte kann allerdings ausgeschlossen werden, dass ein Fluss von der Markierung L über das Umland von A nach O fließt. Hier ist offensichtlich in den Ausgangsdaten ein falsches Apollonia (automatisch) mit der Toponym-Annotation assoziiert worden. Im Graphenmodell kann diese Zuweisung leicht entfernt werden, so dass letztlich der Fluss „Chon“ (korrekterweise) provisorisch zwischen Lacmon und Oricum verortet wird. Die interpretationsabhängige Untersuchung, ob es sich bei Chon angesichts dieser Lage z. B. um den auch als „Aous“ bekannten Fluss mit der Pleiades-ID 481726 handelt, kann nun von fachwissenschaftlicher Seite erfolgen. Ein stets vorsichtiger und kritischer Blick auf die kompakt angezeigten Ergebnisse von Textanalysen ist in jedem Fall gefragt.

Die hier vorgestellten Methoden und das gezeigte Werkzeug sind nicht auf den Spezialfall des **HERODOT-Korpus** beschränkt. Jeder entsprechend annotierte Text kann auf diese Weise verarbeitet und visualisiert werden. Von der Pelagios-Initiative wird derzeit mit Recogito¹² ein kollaboratives Onlinewerkzeug zur geo-zentrischen Textannota-

¹¹Pleiades-IDs 216706, 530972 und 481939

¹²<http://recogito.pelagios.org/>

tion entwickelt. Über dessen [API](#)-Funktionalität ist es möglich, auf die manuell oder semi-automatisch annotierten Dokumente zuzugreifen. Über solche Vernetzungen von Werkzeugen lassen sich künftig auch *Crowdsourcing*-Ansätze verfolgen, in denen einfache Annotationsaufgaben von vielen Freiwilligen erledigt werden, wodurch auch Einzelfehler durch Mehrfachbearbeitung reduziert werden können.

Grundsätzlich existieren jedoch noch offene Fragestellungen beim automatisierten Umgang mit geographischen Entitäten, speziell auch solchen im historischen Kontext. Zum einen ist in bisherigen Ansätzen zur [LOD](#)-Repräsentation von Orten nur wenig einheitliche Modellierung der Granularität der Entitäten vorhanden. So können Werkzeuge meist nur Punktkoordinaten verwenden, was bei Entitäten, die Kontinente, Länder und Provinzen beschreiben, nicht zweckmäßig und visuell unpassend ist. Auch zeitliche Einordnungen und deren Granularitäten – ab wann existieren Orte, ab wann haben sie welche Ausdehnung und tragen sie welchen Namen – sind derzeit nicht ausreichend berücksichtigt. Diese Probleme grenzen an viel weiterreichende Fragen der fachwissenschaftlichen Modellbildung.

Letztlich ist grundsätzlich zu klären, wie heutzutage die Erschließung von Raumbegriffen stattfinden kann, die im Kontext vergangener Zeiten und Kulturen verwendet wurden. Schon die Abbildung auf den kartesischen Raum verändert den Kontext enorm, da genaue Kenntnisse über diesen Raum und seine Abbildung erst in der Moderne aufkamen, Raum also nicht so abgebildet wird, wie er ehemals wahrgenommen wurde. In jedem Fall muss genau geklärt werden, wie historische Texte Ortsbezüge verwenden. Hier sind einfache statistische Methoden sehr auf die Aufdeckung generischer Abhängigkeiten limitiert. An die Stelle der hier vorgestellten Kookkurrenzbetrachtungen können jedoch stets auch Ergebnisse entsprechend optimierter Relationsextraktionsverfahren oder manueller Annotation und Modellierung treten.

4.2.3 Eigennamenübersetzung aus lokal alignierten Paralleltexten

Im [HERODOT-Korpus](#) liegt neben der bis hierhin verwendeten geo-annotierten englischen Übersetzung der Historien auch eine Edition des altgriechischen Textes vor. Diese verfügt im [TEI](#)-Markup allerdings über keinerlei Auszeichnung von Eigennamen. Im Folgenden wird gezeigt, wie die Orts-Annotationen der englischen Version auch für die Behandlung von Toponymen im griechischen Text genutzt werden können. Über das vorgestellte Verfahren kann der verwendete Gazetteer auch um weitere griechische Entsprechungen

für die enthaltenen Ortsnamen erweitert werden.

Die strukturellen Aggregationsmöglichkeiten in der Graphdatenbank, insbesondere der abschnittswisen Alignierung der beiden Sprachvarianten, werden im Folgenden mit lexikalischen Vergleichsoperationen kombiniert. Für das Verfahren werden dabei einige Annahmen getroffen:

- Die Großschreibung eines Wortes in der griechischen Edition wurde vom Editor hauptsächlich zur Hervorhebung von Eigennamen verwendet.
- In der Übersetzung wurde Koreferenz für Entitäten – über Pronomen oder andere Hilfsmittel – nicht quantitativ oder qualitativ wesentlich anders als im griechischen Text verwendet.
- Die Übersetzung ist abschnittsweise inhaltstreu und das sprachübergreifende Vorkommen von Entitäten (und damit den originalen und übersetzten Eigennamen) in den gleichen Abschnitten ist statistisch auffällig.
- Die übliche Übersetzungspraxis für Ortsnamen orientiert sich an der phonetischen Sequenz, welche mit lexikalischen Merkmalen korreliert, welche sich wiederum durch Transliterationsverfahren alphabetübergreifend zu großen Teilen bewahren lassen.

Diese Annahmen sind sicher nicht für alle Übersetzungen korrekt. Im hier beschriebenen Fall scheinen sie aber hinreichend zutreffend zu sein.

Für die Transliterierung wird im Folgenden die Unicode-Funktionalität der [International Components for Unicode \(ICU\)](http://icu-project.org/), einer Werkzeugsammlung für den Umgang mit unicodebasierten Zeichenketten in verschiedenen Schriftsystemen, verwendet. Die in Java geschriebene „Quasi-Referenzimplementierung“ für Unicode-Operationen, die ICU4J-Bibliothek¹³, lässt sich einfach in das System integrieren (und wird auch für die Zeichennormalisierung in Kadmos genutzt). Der folgende als Zeichenkette übergebene Befehl kann verwendet werden, um eine optimierte Transliterators-Instanz zu erzeugen, deren `transliterate`-Methode eine gegebene griechische Zeichenkette in eine vereinfachte Form im lateinischen Alphabet umwandeln kann:

```
1 nfd;[:nonspacing mark:]remove;[:modifier letter:]remove:greek-latin
```

¹³<http://icu-project.org/>

Zunächst wird in die [NFD](#) konvertiert, dann werden kombinierende und modifizierende Diakritika (und ähnliche unerwünschte Zeichen) entfernt und schließlich die Standard-Transliteration vom griechischen ins lateinische Alphabet durchgeführt. Beispielsweise wird aus dem griechischen Χαλκιδέες die Zeichenkette Chalkidees gebildet. Diese weist eine gewisse Ähnlichkeit mit der englischen Übersetzung *Chalcis* auf (im deutschen analog „Chalkida“ oder „Chalkis“).

Zur Bestimmung geeigneter Übersetzungskandidaten im gesamten Dokument genügt dann bereits ein sehr einfacher Ansatz zur Kombination von Erkenntnissen aus Auftretenshäufung und Wortähnlichkeit: Für jedes Toponym t in der englischen Version wird die Menge S_E aller Abschnitte betrachtet, in denen es enthalten ist. In der damit korrespondierenden Menge S_G der selben Abschnitte in der griechischen Version werden alle vorkommenden großgeschriebenen Wörter c in der Kandidatenmenge C_G erfasst. Für diese Kandidaten wird eine Scoring-Funktion wie folgt definiert:

$$\text{score}_t(c) = \frac{\max(l(t), l(c)) - \text{edit}(t, \lambda(c))}{l(t) + l(c)} \cdot \alpha + \frac{\text{co}_t(c)}{\max_{d \in C_G} (\text{co}_t(d)) + |S_E|}$$

wobei die Funktion l die Zeichenlänge angibt, λ die oben beschriebene Transliteration bezeichnet und edit ein Maß für die Unähnlichkeit zweier Wörter ist¹⁴. Es wird zur Berechnung geeigneter Kandidaten also nur ein (im Bezug auf t) lokaler Ausschnitt des Korpus zur Berechnung der Toponymentsprechungen herangezogen.

Für alle $c \in C_G$ wird $\text{score}_t(c)$ berechnet und zur absteigenden Sortierung der Kandidaten genutzt. Zur Bildung eines Schwellwertes für den Score wird zunächst der *Maximalscore* $\text{maxscore}_t = \max_{d \in C_G} \text{score}_t(d)$ ermittelt. Alle Kandidaten die einen höheren Score als einen Schwellwert $\text{maxscore}_t \cdot \beta$ besitzen, werden als Ergebnisliste für t zurückgegeben.

In den folgenden Beispielen wurden die Parameter auf $\alpha = 4$ und $\beta = \frac{3}{4}$ festgelegt. Diese sind (wie im Übrigen das gesamte Verfahren) nicht systematisch optimiert worden, sondern wurden anhand vorangehender Testläufe abgeschätzt. Für das häufig vorkommende *Egypt* wurden dabei die folgenden 18 Entsprechungen zurückgegeben:

Αἰγύπτω Αἰγύπτου Αἴγυπτον Αἰγύπτω Αἴγυπτος Αἴγυπτόν Αἰγυπτίω Αἴγυπτους Αἴγυπτος
Αἰγυπτίη Αἰγύπτιοι Αἰγυπτίων Αἰγυπτίησι Αἰγύπτιον Αἰγύπτια Αἰγυπτίου Αἰγύπιος Αἰγυπτίοισι

¹⁴Es handelt sich dabei um die Levenshtein-Distanz, auf die in Abschnitt [4.5 auf Seite 173](#) noch weiter eingegangen wird.

Hierbei handelt es sich um verschiedene grammatikalische Varianten des korrekten Toponyms, die durchaus für eine Erweiterung des Gazetteers interessant sein können. Für Troja (*Troy*) wurden Τροίης und Ἰλίων als Kandidaten gefunden, letztere trotz großer lexikalischer Unterschiede aufgrund der sehr ähnlichen Muster im Auftreten. Es ist also durchaus ein gemischter Einfluss beider Ansätze für den Wortabgleich zu erkennen.

Die quantitative Auswertung wurde dankenswerterweise unterstützt durch Dr. Elton Barker¹⁵. Es ergab sich in Anbetracht der Einfachheit des Ansatzes ein recht positives Ergebnis: Nur bei 31 der 729 Toponyme (4,25%) war der Übersetzungskandidat mit dem höchsten Score keine korrekte Entsprechung der englischen Bezeichnung. In allen anderen Fällen wurde eine passende Wortform gefunden.

Einige der Negativbeispiele erfordern dabei auch bei manueller Bearbeitung ein gewisses Hintergrundwissen: In der englischen Toponymliste findet sich so z. B. als Benennung eines Felsens der Begriff „*Blackbuttocks*“, nach einem Beinamen für Herkules¹⁶. Das seltene Vorkommen dieses Worts und die nicht an der griechischen Oberflächenform orientierte inhaltliche Übersetzung erschweren eine automatische Zuweisung des korrekten Übersetzungskandidaten ohne zusätzliche Wissensbasis enorm. Die zurückgegebenen Kandidaten sortiert nach Score lauten:

Ἀλφεινὸν Μελαμπύγου Ἀσωπουᾶ Κερκώπων Λοκρίδων

wobei der korrekte Kandidat immerhin an zweiter Position steht.

Das hier vorgestellte Verfahren bietet viele Ansatzpunkte für Erweiterung, Anpassung und eine (über das *Parameter-Tuning* hinausgehende) Optimierung. Dazu lassen sich viele verschiedenen Ideen entwickeln:

Für nicht stellengetreu alignierte Paralleltexte lässt sich eine Vergrößerung des Kontexts um angrenzende Abschnitte durchführen. In der Folge könnte sogar ein zumindest semi-automatischer Alignierungsmechanismus über die gefundenen Entitäten implementiert werden.

Wenn generell auch nicht-lokale Abfragen in das Verfahren einfließen sollen, so könnte beispielsweise in Anlehnung an Tf-idf-Berechnungen auch ermittelt werden, in wie vielen Abschnitten ein Übersetzungskandidat überhaupt vorkommt. Ist diese Zahl deutlich größer als $|S_E|$, so ist es eher unwahrscheinlich, dass es sich um eine korrekte Übersetzung

¹⁵Dozent der Klassischen Altertumswissenschaft an der Open University (Milton Keynes, GB)

¹⁶vgl. „Schwarzsteiß“ in <http://www.vollmer-mythologie.de/melampygos/>

handelt. Zusätzlich könnten auch Übersetzungskandidaten für Anthroponyme und Ethnonyme¹⁷ auf die oben beschriebene Weise berechnet werden. Dann könnten die starken Kandidaten der Ergebnislisten dieser Entitätentypen genutzt werden, um schwächere Kandidaten der Toponym-Ergebnislisten herauszufiltern.

Schließlich bietet auch die Berechnung der lexikalischen Ähnlichkeit noch Verbesserungsmöglichkeiten. In Anbetracht der zahlreichen Wortvarianten, die sich hauptsächlich im Suffix unterscheiden, könnte die Editierdistanz z. B. noch so angepasst werden, dass Unterschiede am Wortende geringere Distanzen verursachen als solche, die zu Beginn der Worte auftreten. Auch häufig auftretende äquivalente Lautentsprechungen sollten geringe Kosten aufweisen, wie beispielsweise beim κ , welches im Englischen meist mit einem c nachgebildet, aber im Programm mit k transliteriert wird.

Auch für diesen Anwendungsfall lässt sich feststellen, dass die Modellierung und Abfrage über Graphdatenbanken die nötige Flexibilität für initiales *Prototyping*, inkrementelle Verbesserung und perspektivische Verfahrenserweiterung bietet.

4.3 Systematisierung und Filterung bibliographischer Daten

Mit dem nun folgenden Anwendungsfall wird sowohl die forschungsgeleitete Integration bibliographischer und biographischer (Meta)-Datensammlungen in ein Property-Graph-Modell demonstriert, als auch der interdisziplinäre Prozess beleuchtet, in welchem die domänen- und fachfragenspezifische Eingrenzung und Auswertung des Datenbestandes umgesetzt wurde. Bei den in diesem Abschnitt vorgestellten Arbeiten handelt es sich um ein Kooperationsprojekt mit Ninja Steinbach-Hüther¹⁸. Sie hat die Hauptarbeit bei der Formulierung der fachspezifischen Forschungsfragen geleistet, sämtliche manuellen Arbeitsschritte und qualitativen Analysen konzipiert und ausgeführt sowie paritätisch die hier beschriebene Forschungsmethodik mitentwickelt. Die technologische Umsetzung

¹⁷Personennamen sowie Benennungen von Völkern und Stämmen

¹⁸Ninja Steinbach-Hüther schreibt ihre Dissertation an der Universität Leipzig und der École normale supérieure, Paris, zum Thema „Zirkulationsweisen afrikanischen Wissens. Präsenz und Rezeption akademischer Literatur aus Afrika in Deutschland und Frankreich“ und ist Wissenschaftlerin im Sonderforschungsbereich 1199, „Verräumlichungsprozesse unter Globalisierungsbedingungen“, Teilprojekt C1 (Universität Leipzig / Leibniz-Institut für Länderkunde (IfL), Leipzig).

<http://research.uni-leipzig.de/~sfb1199/>

für Datenimport, -modellierung, -transformation, -filterung und -visualisierung wurde dagegen eigenständig vom Autor dieses Buches vorgenommen.

Das Projekt beschäftigt sich mit Fragen und Problemen des Kulturtransfers und lässt sich inhaltlich in den „*Global and Area Studies*“ verorten. Ausgangspunkt ist die Frage, inwieweit in den Sozial- und Geisteswissenschaften seit der Mitte des letzten Jahrhunderts an der globalen Wissensproduktion afrikanische Autoren partizipieren. Dabei wird der Umgang mit und die Rezeption von entsprechenden Veröffentlichungen (hier speziell Monographien) in nicht-afrikanischen Ländern als Untersuchungsgegenstand gewählt. Das Forschungsumfeld und vorangehende Arbeiten zu diesem Thema werden in [ESH14] detaillierter vorgestellt. Dabei werden auch die Grundzüge des Projekts genauer motiviert und dokumentiert, als es hier möglich ist.

In überwiegendem Maße begegnen die bisherigen Forschungsansätze dem Thema auf exemplarische Art und Weise durch tiefgreifende qualitative Betrachtungen kleiner Ausschnitte. So wird z. B. die Wirkhistorie einzelner Autoren, Verlage und anderer individueller Akteure nachgezeichnet. Eine repräsentative „Bestandsaufnahme“ relevanter Veröffentlichungen aus dem afrikanischen akademischen Umfeld findet dabei allenfalls für sehr kleine thematische Teilgebiete statt.

Das Ziel des hier vorgestellten Kooperationsprojekts ist es, komplementär zu diesen punktuellen (und im weitesten Sinne dem *close reading* zuzuordnenden) traditionellen Analysen eine bisher nicht berücksichtigte quantitative Sichtweise auf der Basis großer digital vorliegender Datensammlungen zu entwickeln. Damit lassen sich Entwicklungen und Zusammenhänge aus einer „Makro-Perspektive“ betrachten, wodurch sowohl grundsätzliche Aussagen über die Zusammensetzung und Dynamik der Rezeption afrikanischer Wissensproduktion getroffen als auch lohnenswerte Ansatzpunkte für weitere zielgerichtete qualitative Untersuchungen gefunden werden können.

Die Untersuchung stützt sich auf eine umfassende Sammlung bibliographischer Daten, die durch die Französische Nationalbibliothek, die [Bibliothèque nationale de France \(BnF\)](#), erfasst wurden. Die [BnF](#) hat einen öffentlichen Sammelauftrag, der alle französischsprachigen sowie alle in Frankreich verlegten Publikationen einschließt. Der Datensatz ermöglicht einen Einblick in den französischen Buchmarkt und die sich dort widerspiegelnde Sichtbarkeit afrikanischer Autoren im Bereich sozial- und geisteswissenschaftlicher Forschungsthemen.

Die Rezeption der Resultate afrikanischer Forschungstätigkeit außerhalb ihrer Herkunftsländer wird dabei durch die Linse eines Landes mit ausgeprägter kolonialer Vergangenheit und sprachlichen Verbandelungen auf den afrikanischen Kontinent gesehen, welches häufig zugleich Wirkungsstandort afrikanischer Akademiker ist. Eine Kontrastierung der im Folgenden vorgestellten auf [BnF](#)-Daten basierenden Analysen mit Ergebnissen aus (bisher hauptsächlich manuell ausgewerteten) Katalogdaten der Deutschen Nationalbibliothek findet im Rahmen der fachwissenschaftlichen Ergebnisinterpretation statt und soll in diesem Buch daher ausgeklammert werden.

Die Daten wurden direkt von der [BnF](#) bezogen und von der Fakultät für Sozialwissenschaften und Philosophie der Universität Leipzig für die Untersuchung weiterführender Forschungsfragen erworben. Dabei wurde eine Vorselektion vorgenommen, auf nach 1950 in Frankreich verlegte Bücher mit mindestens einem zugeordneten Autor, welchem in den Stammdaten der [BnF](#) („*catalogue général*“) ein afrikanischer Länder- oder Sprachencode zugewiesen ist. Unter Beachtung dieser Bedingungen wurden korrespondierende Datensätze mit 51266 Autoreneinträgen und 64574 Büchereinträgen extrahiert und im Dateiformat des Tabellenkalkulationsprogramms Excel zur Verfügung gestellt. Die Spalten orientieren sich dabei am [BnF](#)-eigenen [MARC](#)-basierten Metadatenformat „*Intermarc*“. Alle Einträge besitzen eine eindeutige Identifikationsnummer. Zum Teil sind in Feldinhalten bestimmte Zeichenketten, wie „\$a“ enthalten, die eine Separierung in mehrere Teilwerte anzeigen. Solche Multi-Wert-Felder werden z. B. für die Verknüpfung von Büchern mit Autoren-IDs verwendet.

Die Datensätze zu Büchern beziehen dabei nicht einzelne Exemplare (*Items*) ein, etwa mit Informationen zur Regaleinordnung, wie sie in vielen Bibliothekssystemen üblich sind. Der vorliegende Datensatz bewegt sich auf der Ebene der *Manifestation*, also physisch in Serie erzeugter Editionen und damit Ausgaben von verschriftlichten (in Form einer *Expression* realisierten) Werken (*Works*) – die Unterscheidung in diese konzeptuellen Ebenen in Bibliotheksdatensätzen wurde mit dem [Functional Requirements for Bibliographic Records \(FRBR\)](#) eingeführt.

Die adäquate informatische Behandlung eines solchen Forschungsszenarios erfordert insbesondere die Nutzung (oder Erstellung) eines geeigneten Werkzeugs zur Systematisierung und Filterung der Daten in einer Form, in der die Beantwortung komplexer Forschungsfragen möglich ist. Diese (teils sehr konkreten) Forschungsfragen sind stark datenbezogen und zu Beginn des Projekts daher noch nicht im Detail bekannt. Sie ergeben sich schrittweise während der explorativen Beschäftigung mit den Datensätzen und den

daraus ableitbaren Mustern im Vergleich zu bestehenden Theorien.

Für ein solches Forschungsszenario ist die Nutzung wissenschaftlicher Bibliographie-Verwaltungs-Software, welche oft den Bereich des persönlichen Wissensmanagements adressiert, nicht ausreichend. Die dort vorhandenen Werkzeuge orientieren sich oft an Modellierungskonstrukten von $\text{BIB}\text{T}_{\text{E}}\text{X}$ ¹⁹, was die Verwaltung der Daten schon für einfachste quantitative Betrachtungen sehr erschwert. In [Ley09] wird diese Problematik näher beschrieben. Unter anderem ist die Berücksichtigung der genauen Identität von Autoren dort meist nicht gegeben, da Autorenschaft üblicherweise nicht über die Namensnennung hinaus abgebildet werden kann.

Im Bereich der Szientometrie und deren Unterdisziplin, der Bibliometrie, werden zwar quantitative Untersuchungen von wissenschaftlichen Veröffentlichungen und deren Autoren durchgeführt, der Fokus liegt dort jedoch fernab des hier betrachteten Kulturtransfers. Die üblichen Analysen von Zitation (und Koautorenschaft) sind hier nicht möglich bzw. nicht zielführend (wie später noch gezeigt wird). Entsprechend verfügen auch die einschlägigen Werkzeuge, wie SciMAT²⁰ und Sci2²¹ nicht über alle der hier benötigten Funktionen.

Auch generische Werkzeuge zur Datenaufbereitung und Informationsintegration, einem Bereich, der meist mit **Extract, Transform, Load (ETL)** bezeichnet wird, bieten sich nicht für eine Nachnutzung an: Open Refine²² bietet zwar umfangreiche Möglichkeiten zur Datenfilterung, -konvertierung und -vereinheitlichung, besitzt allerdings den Nachteil, dass es Eingangsgrößen als reine tabellarische Datensätze auffasst. Die Datengrundlage in diesem Projekt basiert zwar auch auf Tabellen, jedoch sind diese über Identifikationsnummern verknüpft und genau diese Verknüpfung soll als Ausgangsbasis für Filterungsschritte dienen. Hinzu kommt, dass Open Refine zwar nützliche Operationen, wie Datennormalisierung und das Aufsplitten und Zusammenführen von Feldinhalten unterstützt, diese aber irreversibel und permanent sind – mit Ausnahme einer linearen *Undo/Redo*-Funktion. Beim dort unterstützten Arbeitsablauf fehlt jede Möglichkeit des Experimentierens mit und Dokumentierens von einzelnen Verarbeitungsschritten. Werkzeuge für die Schaffung visueller Zugänge zu multidimensionalen Daten durch interaktive Analysen sind ebenfalls meist tabellenorientiert und zudem meist nicht mit genügend

¹⁹Populäres Literaturverwaltungssystem für den Einsatz mit dem Textsatzsystem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

²⁰<http://sci2s.ugr.es/scimat>

²¹<http://sci2.cns.iu.edu/user/index.php>

²²<http://openrefine.org/>

Transformationsmöglichkeiten ausgestattet – so z. B. DataComb²³, wo Ideen aus [RC94] implementiert werden, RAW²⁴ oder Brunel²⁵.

Werkzeuge für die Metadatenprozessierung im Bibliothekssektor, wie beispielsweise Metafactory²⁶ mit der Transformationssprache Metamorph, s. [BGH15], und einer eigenen Scriptsprache namens „Flux“ bieten dagegen gute Transformationsmöglichkeiten, haben jedoch üblicherweise wiederum einen komplett anderen Fokus, nämlich die Definition statischer Transformationsprozessketten und Formatkonverter für veränderliche Datensätze, wohingegen hier flexibel veränderbare Filterungs- und Transformationsmöglichkeiten für einen statischen Datensatz benötigt werden.

Aus der Durchsicht all dieser jeweils nur partiell geeigneter Werkzeuge ergibt sich die Notwendigkeit der Entwicklung eines eigenen Systems, welches auf der Basis verknüpfter Daten in einem Property-Graph-Modell arbeiten soll. Über das Ruby-Gem Roo²⁷ kann ein Import der Daten leicht erfolgen, wobei Multi-Wert-Felder bereits an dieser Stelle aufgesplittet und ihre Einzelwerte in Knoten umgewandelt werden. Die Verknüpfung der Basisinformationstypen „Buch“ und „Autor“ hat dabei die Besonderheit, dass der Datensatz unterschiedliche Formen der Autorenschaft angibt. Diese können im Property-Graph-Modell allerdings einfach als Kantentypen repräsentiert werden. Bei der BnF werden nach InterMarc-Kodierungsschema 209 Rollen unterschieden, in denen die Mitwirkung von Personen an Buchpublikationen beschrieben werden kann. Diese „Autorenrollen“²⁸ umfassen Standardrollen, wie den Textautor (0070 „Auteur du texte“), aber auch sehr spezifische Funktionen, wie Erfinder, Kopist, Illustrator oder Vorwortschreiber. Abbildung 4.7 auf der nächsten Seite zeigt schematisch, dass nur bestimmte Rollen (und damit nur bestimmte Gegenspieler bei Autorenschafts-Kanten) für die Betrachtungen relevant sind.

Mit den so verknüpften Daten ist es möglich, erste visuelle Auswertungen des Datensatzes vorzunehmen. Die den Autoren zugewiesenen Länder können später genutzt werden, um einen geographischen Bezug der einzelnen als relevant erachteten Bücher herzustellen. Dafür muss jedoch zunächst die Datengrundlage genauer untersucht werden. Abbildung 4.8 auf Seite 155 zeigt die Aggregation von ca. 39000 Ländereinträgen der

²³<http://www.bytemuse.com/post/data-comb-visualization>

²⁴<http://app.raw.densitydesign.org/>

²⁵http://brunel.mybluemix.net/gallery_app/renderer

²⁶<http://culturegraph.github.io/>

²⁷<http://github.com/roo-rb/roo>

²⁸http://www.bnf.fr/documents/intermarc_ref_fonctions.pdf

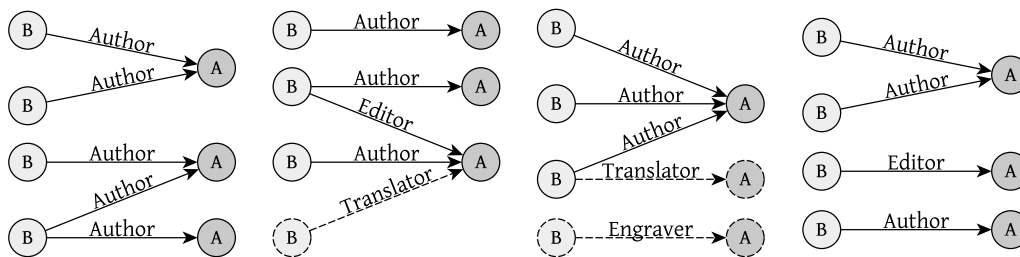


Abbildung 4.7: Skizze zur Verknüpfung von Büchern und Autoren in verschiedenen Rollen im Datenmodell

in den Ausgangsdaten enthaltenen Autoren. Es ist zu erkennen, dass auch Länder außerhalb Afrikas enthalten sind. Der Grund dafür ist, dass manchen Autoren mehrere Länder zugewiesen sind und andere wiederum durch ihre Kenntnis einer afrikanischen Sprache in der Datenbasis enthalten sind, aber im Datensatz eine nicht-afrikanische Staatsbürgerschaft zugewiesen bekommen haben. Später müssen verbliebene Autoren dieser Konfiguration einzeln manuell überprüft werden, um sicherzustellen, dass sie der fachwissenschaftlich gewünschten Definition „afrikanischer“ Autoren im Kontext der Forschungsfrage genügen.

Die Visualisierung kann für direkte Länderangaben (Autoren) aber auch indirekt ermittelbare (Bücher) und abfragespezifische Ausgaben (z. B. „Koautoren ägyptischer Autoren mit mehr als 5 Veröffentlichungen“) verwendet werden. Sie kann ebenso beliebige Filterungsstände reflektieren, die dann visuell untereinander verglichen werden können. Die intensivste Färbung repräsentiert dabei jeweils das (absolut) am häufigsten enthaltene Land, Schattierungen davon bedeuten abgestufte Anteile. Die Werte sind nicht nach Fläche, Einwohnerzahl oder anderen Merkmalen der Länder normalisiert, haben also keinen direkten Flächenbezug und nutzen die Karte nur als geographische Orientierung zur Verortung der Länder²⁹. Aus der Datenbasis können für diese Darstellungsform leicht Extrakte einzelner Zeitabschnitte (auf Basis der Jahresangaben) gebildet werden, welche dann z.B. auch in Form einer Animation Aufschluss über die temporale Entwicklung geben können.

²⁹Selbst nach der Anwendung theoretisch abgeleiteter und thematisch angemessener Normalisierungsmaße wäre nicht unbedingt eine der statistischen Eigenschaften der geographischen Verteilung angemessene Interpretierbarkeit der Karte gegeben, wie z. B. in [BDM⁺17] thematisiert wird. Überdies wird auch auf eine Differenzierung der diachron veränderlichen Staatsaufteilung verzichtet, was sich z. B. in der Ausblendung von Somaliland und dem Südsudan zeigt, in anderen Ländern aber zu leichten Verzerrungen führt. Insgesamt ist das Ziel nur die Ermöglichung eines leichten Auffindens der am stärksten präsenten Länder und deren Abgrenzung zu sehr schwach repräsentierten Ländern.

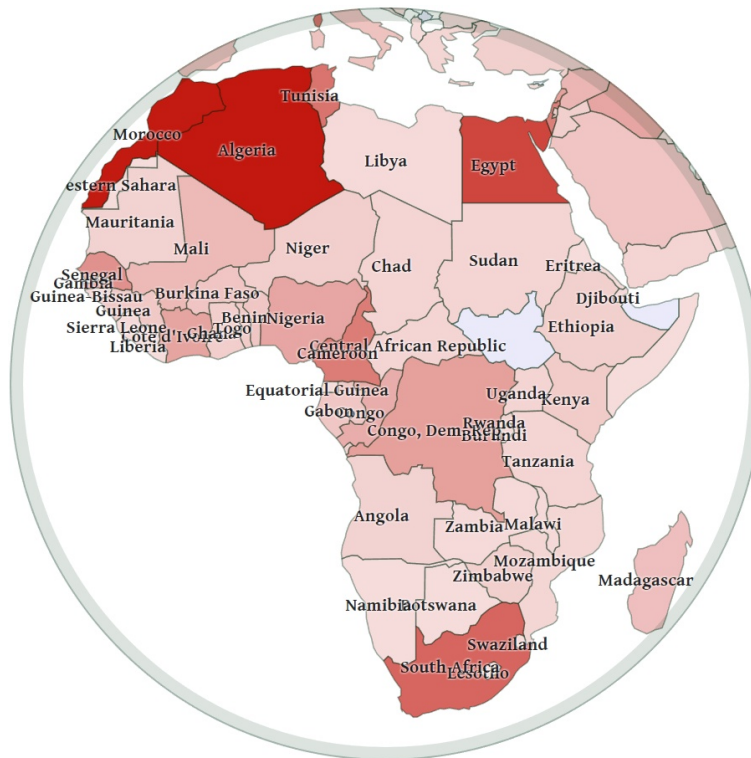


Abbildung 4.8: Kartogramm-ähnliche Darstellung der Autorenverteilung nach Land im Ausgangsdatenbestand

Die entwickelten Visualisierungen, von denen hier nur eine Auswahl gezeigt werden kann, sind browserbasiert und wurden unter Verwendung der in Javascript geschriebenen Programmbibliothek D3 erstellt, vgl. [BOH11]. Es zeigt sich in vielen Anwendungsfällen, dass Visualisierungstechniken bewusst und defensiv eingesetzt werden müssen. Aus einer rein relativen Angabe der Geschlechterverhältnisse wie in [Abbildung 4.9 auf der nächsten Seite](#) oben gezeigt, könnte z. B. voreilig abgeleitet werden, dass sich seit den 1950er Jahren bis in die 1980er Jahre der Anteil weiblicher Autoren schubweise von 0% auf 10% erhöht hat. Angesichts der „River“-Darstellung unten wird dabei jedoch deutlich, dass in der Anfangszeit im Schnitt nur fünf bis zwanzig Bücher gelistet sind³⁰ und deshalb Aussagen zu relativen Verhältnissen dort nicht auf Basis einer verlässlichen Grundgesamtheit getätigt werden können.

Nach initialen Visualisierungen zur Verschaffung eines Überblicks über die Daten, müssen nun Facetten geschaffen werden, durch die es gelingt, effizient zu filtern und zu aggregieren. Eine der erweiterten Aufgaben des [ETL-Prozesses](#) ist in diesem Zusammen-

³⁰Beim Überfahren der Visualisierung mit der Maus werden die konkreten Werte für das jeweilige Jahr angezeigt

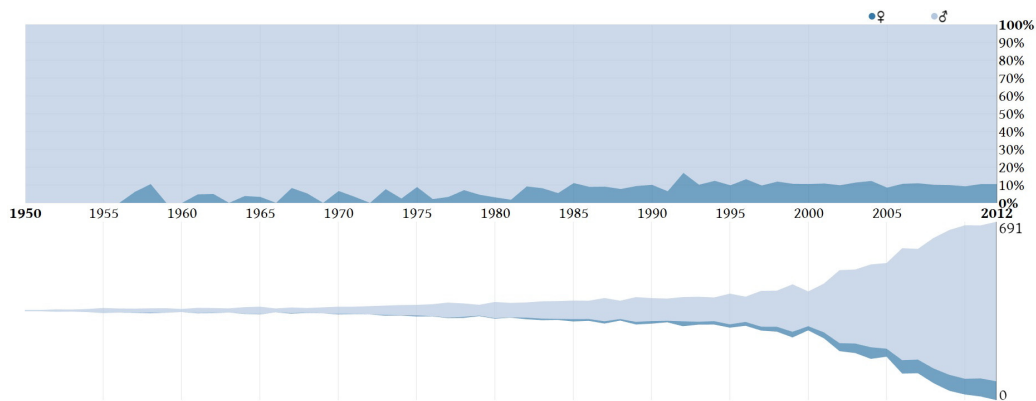


Abbildung 4.9: Relativer Anteil eingetragener Geschlechter für Buchautoren über die Zeit der Buchpublikationen – Prozentsatz und Stream-Ansicht absoluter Verhältnisse

hang die Datenbereinigung hinsichtlich unterschiedlicher Benennungen von Entitäten. Dies umfasst zum Teil auch semi-automatische Schritte, um z.B. Einträge im Feld für Verlagsangaben korrekt auf eine Basisbezeichnung zurückzuführen. Entsprechend müssen geeignete Zeichenketten gefunden werden, die für eine Suche von Entitäten verwendet werden können. Hierbei wurde z. B. „armat“ als geeigneter Substring identifiziert, um Verknüpfungen zu einem neu angelegten Knoten für den Verlag „L'Harmattan“ durchzuführen. Die einzelnen Einträge mit diesem Substring beinhalteten die folgenden Verlagsangaben:

l'Harmattan d.l'Harmattan Ed.l'Harmattan l'Harmattan-Sénégal l'Harmattan-Cameroun l'Harmattan-RDC ditions l'Harmattan l'Harmattan-Congo L'Harmattan Academia-l'Harmattan l'Harmattan Sénégal l'Harmattan Cameroun l'Harmattan-Cameroun d. L'Harmattan diff. l'Harmattan l'Harmattan-[Côte d'Ivoire l'Harmattan Congo l' Harmattan Ed. L'Harmattan l'Harmattan Italia l'Harmattan-Côte d'Ivoire d.l'Harmattan l'Harmattan l'Harmattan-[Congo le Scribe l'Harmattan l'Harmattan-[Sénégal Ed l'Harmattan Espace l'Harmattan Kinshasa l'Harmattan Gabon l'Harmattan Mauritanie l'Harmattan-ACIVA

Wenn das Netzwerk entsprechend mit extrahierten Entitäten angereichert ist, kann mit der Filterung des Datenbestandes begonnen werden. Allerdings genügen die aus der Struktur ablesbaren Muster nicht, um die Eingrenzung der Untersuchungsgegenstände entsprechend der Forschungsfrage vorzunehmen. Als wichtige zusätzliche Aufgabe ist die Filterung nach thematischem Fokus eines jeden Buches anzusehen. Im Einzelfall ist die Entscheidung über Relevanz dabei alles andere als trivial und nicht mit der nötigen Präzision auf automatischem Wege zu treffen. Vereinfacht lässt sich diese Thematik z. B. so illustrieren, dass Bücher über ingenieurtechnische Aspekte des Staudammbaus nicht von Interesse sind, wohingegen Bücher über die sozialen Folgen des Staudammbaus sehr wohl Betrachtung finden sollten.

In den meisten Bibliotheken findet eine Kategorisierung des Bestandes nach Themen statt, so auch bei der [BnF](#), wo dafür die Dezimalklassifikation nach Dewey genutzt wird.³¹ Dabei handelt es sich um ein hierarchisches System, bei dem die erste, signifikanteste Stelle die höchste Granularität besitzt. Eine so vereinfachte ontologische Sichtweise ist selbstverständlich nicht ausreichend, um das Thema eines Buches zu beschreiben. Daher werden den entsprechenden, meist sechsstelligen Codes üblicherweise noch andere als Zahlen kodierte Konkretisierungen oder Nennungen von Querschnittsaspekten (wie Ortsbezug) angehängt. Zudem werden zunehmend auch mehrere Kodierungen pro Buch verwendet.

Das große Spektrum möglicher Angaben soll anhand des folgenden Beispiels verdeutlicht werden: Das Buch „*Possession, magie et prophétie en Algérie*“³² besitzt die drei Klassifizierungen:

- 306.089_92765 (21. Dewey-Edition) – mit 11 verknüpften Einträgen,
- 299.613 – „*Culte public et autres pratiques (religions africaines)*“ – mit 13 Treffern und
- 133.408_992765 – „*Démonologie et sorcellerie – Étude en relation avec les Algériens*“ (22. Dewey-Edition) mit nur diesem einem verknüpften Buch

Die Angaben zur Anzahl so verschlagworteter Werke bezieht sich auf den kompletten, ungefilterten Datenbestand der [BnF](#)³³. Aus der hohen Spezifität der einzelnen Einträge und der großen Heterogenität der Kodierungs-Kategorien ergibt sich eine Situation, in der weder ein flächendeckender manueller Ausschluss großer Kategorien noch eine automatisierte Unterstützung, etwa über Verfahren des (überwachten) maschinellen Lernens Aussicht auf Erfolg haben³⁴. Gleichzeitig ist die erneute manuelle Auszeichnung aller Bücher unter Berücksichtigung der Forschungsfrage nicht mit vertretbarem Aufwand umsetzbar – auch, wenn sich daraus viele weitere Analysemöglichkeiten ergeben und insbesondere auch die Entwicklung von Modellen für die Wissensproduktion, wie z. B. von Ginda, Scharnhorst und Börner [[GSB16](#)] beschrieben, möglich wird.

Alternativ wurde eine Kombination aus regelbasierten Filterungsschritten und datengetriebenen Visualisierungen und Listenansichten gewählt, mit der schrittweise ein Ausschluss unerwünschter Datensätze stattfinden kann. Ergänzt wurden diese interak-

³¹s. „la 23e édition de la Classification décimale Dewey“ hier:

<http://asted.org/webdewey-r-en-francais-6688.html>

³²<http://catalogue.bnf.fr/ark:/12148/cb34743618d>

³³Stand: 23.11.2016

³⁴Eine der wenigen Ausnahmen dazu bilden z. B. die Kategorien zu den Naturwissenschaften, vgl. Abbildung [B/4](#) auf [Seite 249](#) im Anhang.

tiven Filterwerkzeuge um eine Stichwortsuche, die zum Ausschluss größerer Mengen eindeutig unpassender Literatur verwendet werden kann. Einzelne Filterungsschritte können dabei jederzeit deaktiviert werden. Dabei werden derzeit noch komplette Neufilterungen mit den verbliebenen Regeln ausgeführt. Dies könnte optimiert werden, wenn sich bei größeren Datensätzen Geschwindigkeitsprobleme ergeben sollten. Die Prozessierung kommt jedoch ohne einen Neu-Import der Daten aus und verwendet nur die Graphen-Repräsentation – ohne Originalwerte zu überschreiben.

- Filtern von Autoren nach
 - Ländercodes,
 - Geburtsjahr (zum Ausschluss von Editionen antiker Autoren),
 - Vorhandensein auf semi-automatisch angelegter Negativliste.
- Filtern von Büchern, bei denen
 - nach obiger Filterung kein Autor (in geeigneter Rolle) verblieben ist,
 - Negativ-Schlagwörter in entsprechenden Feldern (Titel, Kurzbeschreibung, Genre) enthalten sind (um ungewünschte Literaturformen wie z. B. Autobiographien und Reiseberichte zu filtern),
 - bestimmte Verlage verknüpft sind (z. B. zum Filtern dezidierter Jugendliteratur-Verlage),
 - ein Eintrag in der semi-automatisch angelegten Negativliste besteht.
- Filtern von Autoren, denen nun kein Buch mehr (über eine geeignete Rolle) zugeordnet werden kann.

Die so aufbereiteten und gefilterten Daten können in ihrer Graphenstruktur nun grundsätzlich auch mit Mitteln der Netzwerkanalyse untersucht werden. Jedoch zeigt bereits ein oberflächlicher Blick auf die Graphenvisualisierungen in [Abbildung 4.10 auf der nächsten Seite](#), dass beispielsweise das (in ungefilterten Datensätzen zuweilen recht dicht verknüpfte) Koautoren-Netzwerk im Fall des reduzierten Datenbestandes nur wenig zusammenhängend ist.³⁵

Wie eingangs beschrieben, soll hier nicht im Detail auf die fachwissenschaftliche Auswertung der Daten eingegangen werden, sondern primär nur der Weg von den Rohdaten hin zu einer flexiblen Rechercheumgebung für eine präzise und effiziente Abgrenzung und Auswertung des relevanten Materials beschrieben werden.

³⁵Die automatische Erkennung dicht zusammenhängender Teilgraphen für die Abbildung wurde durch das Clusteringverfahren „Chinese Whispers“ umgesetzt, s. [\[Bie06\]](#).

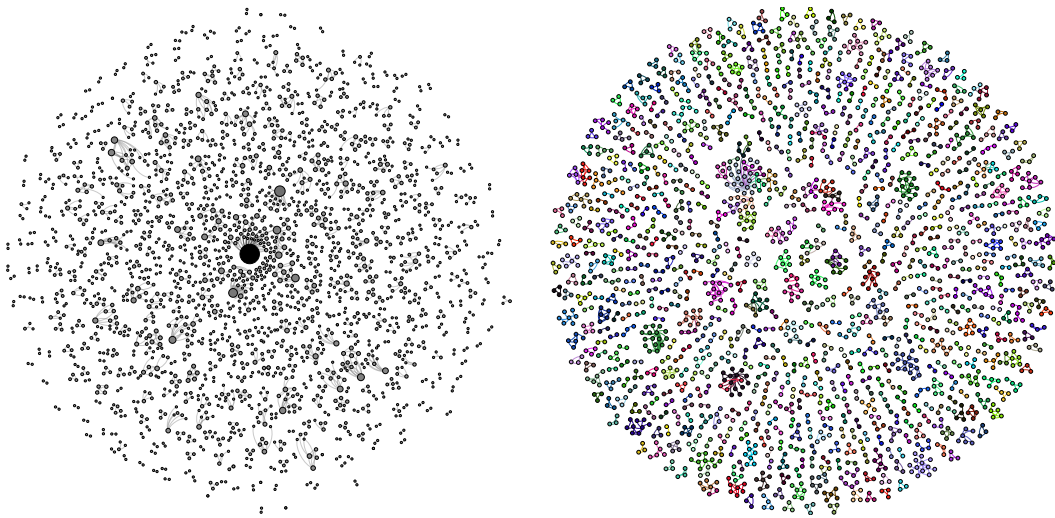


Abbildung 4.10: Visualisierungsvarianten für das Koautoren-Netzwerk in kräftebasiertem Layout in Gephi, links mit Knotengröße entsprechend des Knotengrades und rechts mit Einfärbung nach Knoten-Clustering

Alternativ zum hier beschrittenen Weg wäre denkbar, als Ausgangsbasis auch die von der [BnF](#) bereitgestellten Versionen des Katalogdaten-Gesamtexports im [RDF](#)-Format³⁶ zu verwenden. Der Prozess der Erzeugung dieser Daten wird in [\[SWMM13\]](#) näher beschrieben. Es lässt sich jedoch feststellen, dass die dabei durchgeführte semantische Anreicherung und Verknüpfung einen grundlegend anderen Fokus als die in diesem Projekt durchgeführten Arbeiten besitzt. Zudem bringt der große Umfang der Datensammlung (mit 38 GB [XML](#)-Rohdaten) einen erheblichen technischen Mehraufwand mit sich, vgl. dazu auch [\[ESH14\]](#). Letztlich bleibt auch aus wissenschaftlicher Sicht abzuwägen, ob die integral wichtigen Schritte der Datenintegration und Normalisierung tatsächlich in einem (zwangsweise) intransparenten externen Prozess ablaufen sollten.

Das erstellte Recherchewerkzeug bietet einen angemessen großen Funktionsumfang und besitzt grundsätzlich gute Erweiterbarkeit, sollte sich jedoch nicht als „Datensilo“ und letzte Station für alle notwendigen Analysen positionieren. Daher wurde eine Exportfunktion integriert. Das zum Einlesen der Primärdaten verwendete Ruby-Gem kann auch genutzt werden, um den aktuellen Filterungsstand in Form einer tabellarischen Ansicht für Bücher und Autoren auszugeben. Dies bietet sich an, damit bestehende Arbeitsabläufe, die bisher in Excel umgesetzt wurden, auch weiterhin durchführbar sind.

Mit diesem Anwendungsfall wurde gezeigt, wie ein großer bibliographischer Datenbestand durch die Überführung in eine Property-Graph-Repräsentation und die Erstellung

³⁶<http://data.bnf.fr/semanticweb-en>

leichtgewichtiger Werkzeuge für die Exploration, Filterung, Transformation und Analyse für eine fachwissenschaftliche Auswertung erschlossen werden kann. Dadurch werden Forscher in die Lage versetzt, qualifiziert mit Datenmengen zu operieren, die in konventioneller Arbeitsweise nicht ausgewertet werden könnten.

4.4 Struktur und Komplexität von Dramen

4.4.1 Extraktion, Analyse und Visualisierung von Struktur

Bei dramatischen Werken handelt es sich um vielschichtige szenische Beschreibung der Interaktion von Akteuren, ihren Gesprächen, sonstigen Äußerungen und teils auch Gedanken. Dabei sind viele komplexe Strukturen erkennbar, die sich über den eigentlichen Text legen und welche dabei oft expliziter beschrieben sind, als in anderen Formen der Belletristik. Für eine Analyse ergeben sich daraus viel mehr Ansatzpunkte und zu betrachtende Kontexte, was diese Quellengattung auch zu einem besonders interessanten Untersuchungsgegenstand in den e-Humanities macht.

Der Begriff Komplexität ist dabei intuitiv recht einfach fassbar – Sachverhalte und Zusammenhänge, die sich einer einfachen Beschreibung entziehen, sind komplex. Je mehr Aspekte zur adäquaten Beschreibung zwingend benötigt werden, umso komplexer ist vermutlich das Beschriebene. Solche Komplexitätskriterien lassen sich zwar sehr einfach in natürlicher Sprache formulieren, umso schwerer fällt es aber, sie zu formalisieren oder gar zu berechnen, wie z. B. die Kolmogorow-Komplexität in der Informatik zeigt, s. z. B. [LV09]. In Anbetracht dessen soll die Betrachtung der Komplexität von Dramen hier auch zunächst im informellen Sinne geschehen, bevor sie später noch durch eine mathematische Herangehensweise unterstützt wird.

Zur Einschätzung von Komplexität gehört eine Betrachtung von Regelmäßigkeit und Unregelmäßigkeiten, dem Grad von Verknüpfung und der Intensität von Interaktion innerhalb eines Systems. Die in den e-Humanities entwickelten Technologien sollen helfen, genau solche Aspekte und Zusammenhänge aufzudecken, sie begreifbar, abstrahierbar und vergleichbar zu machen. Daraus ergibt sich auch eine starke Verbindung zu Methoden der Visualisierung – besonders der Informationsvisualisierung und der *Visual Analytics*, wie bereits kurz vorgestellt.

Für die algorithmisch unterstützte quantitative Analyse von Dramen ergibt sich zunächst die Frage nach dem eigentlichen Untersuchungsgegenstand und seiner Repräsentation im digitalen System – mithin die Frage nach einem geeigneten Modell. Dramen sind Texte, weisen aber im Vergleich zu anderen Texten Besonderheiten auf, denn: *„Although plays exist which were mainly written for a reading audience, dramatic texts are generally meant to be transformed into another mode of presentation or medium [...]“* [LM04]

Im Rahmen dieser Arbeit sollen Dramen in Form von Property-Graphen abgebildet werden. Die Frage ist dabei nicht primär, ob die Struktur eines Dramas tatsächlich einem Graphen ähnelt, sondern ob aus einer Graphenrepräsentation eines dramatischen Werks ein geeignetes Modell für die Analyse von Handlungsstrukturen und Ähnlichem abgeleitet werden kann. Wie bereits gezeigt, kann aus einer vernetzten Textrepräsentation auf Tokenebene sehr flexibel zwischen Mikro- und Makroperspektiven gewählt und gewechselt werden. Zudem hilft sie, direkte und indirekte Zusammenhänge zwischen Datensätzen abzufragen.

Tatsächlich ist es die erweiterte Struktur, die Dramen besonders macht. Struktur ist wieder im Sinne von Schachtelung und Abfolge gegeben, es werden jedoch neue Ebenen eingeführt. Regieanweisungen wechseln sich mit Szenenbeschreibungen ab. Charaktere tragen mit An- und Abwesenheit auf der Bühne und verschiedenen Formen von Sprechakten und Handlungen direkter und durch diskretere Strukturen zum Narrativ bei.

Die hier überblicksartig vorgestellten Methoden und Werkzeuge wurden im Rahmen der in [EHJ15] beschriebenen Arbeiten für die Analyse des SHAKESPEARE-Korpus entwickelt. Ähnliche Ansätze – jedoch auf Basis von XML-Technologien – verfolgt die Initiative „Digital Literary Network Analysis“³⁷. Dort wird auch eine etwas andere Form der Graphinduktion gewählt, die anhand der als Property Graph gespeicherten Texte auch in dieser Arbeit umgesetzt werden könnte.

Die hier verwendete Methode betrachtet für den Aufbau des Analysenetzes die Sequenz von Redeakten innerhalb von Szenen. Folgen zwei Redeakte aufeinander, wird zwischen den jeweils sprechenden Charakteren eine Kante angelegt. Bei mehreren solchen Stellen wird das Gewicht der Kante erhöht. Schon Moretti, der ebenfalls Dramenstrukturen (allerdings mutmaßlich händisch) in Graphen überführt hat, spricht sich in [Mor11] für die Nutzung gewichteter Kanten für die Netzwerkanalyse in Dramen aus:

³⁷<http://dlina.github.io/>

[...] when Claudius tells Horatio in the graveyard scene, 'I pray thee, good Horatio, wait upon him', these eight words have in this Figure exactly the same value as the four thousand words exchanged between Hamlet and Horatio

In der automatischen Analyse werden dabei Gruppenbezeichnungen, wie „Die Räuber“, „Beide“ oder „Alle“ nicht aufgelöst und unterschiedliche Benennungen der selben Charaktere, wie „Die Königin“ und „Gertrude“ nicht zusammengeführt. Über entsprechend qualifizierte Annotationen oder Heuristiken lässt sich die Analysequalität noch steigern.

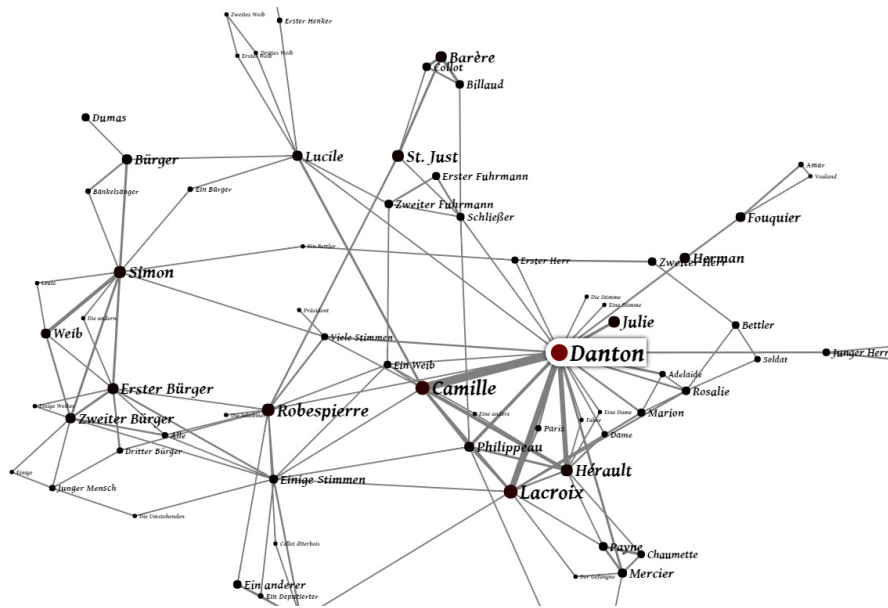
Grundsätzlich stellt sich für eine solche Form der Netzwerkinduktion die Frage, welche Effekte dadurch sichtbar gemacht werden. Bisher wird nur die Abfolge von Redevorgängen dokumentiert. Ob beide Charaktere miteinander sprechen, wird so nicht klar. Deshalb soll hier für die Erzeugung von Kanten ein Schwellwert für die Interaktion von Charakteren eingeführt werden, über den sich regeln lässt, wie oft Charaktere am Stück im Wechsel sprechen (und so vermutlich „antworten“) müssen, damit eine Kante entsteht oder ihr Gewicht inkrementiert wird.

Abbildung 4.11 auf der nächsten Seite zeigt das so erstellte Interaktionsnetzwerk in Georg Büchners „Dantons Tod“ (4 Akte, 32 Szenen) aus dem [DRAMEN-Korpus](#). Für die Visualisierung wurde wieder auf Sigmajs und ForceAtlas2 [\[JVHB14\]](#) zurückgegriffen. Durch das Erhöhen des Schwellwerts zerfällt das Netzwerk langsam in einzelne Komponenten.

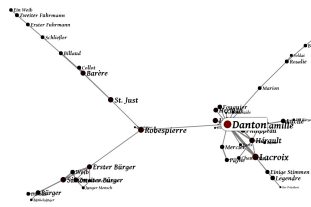
Die Einführung des Dialog-Schwellwerts erlaubt eine Ausblendung von marginaler Interaktion und vermeidet so weitestgehend die optische Gleichsetzung disproportionaler Relationen.

Angeichts des sequenziellen Leseflusses und besonders der sequenziellen Aufführung eines dramatischen Werks ist die Aggregation der Interaktionen des gesamten Textes in eine einzige Darstellung ein Instrument zur Verdichtung, das je nach Interpretationsziel hilfreich oder störend ist. Es existieren valide Argumente gegen ein „Einfrieren“ oder Abstrahieren der Handlung unter Reduktion der sequenziellen Abfolge.

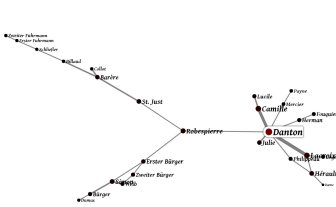
Zur Unterstützung solcher Sichtweisen bietet sich die Ableitung einer alternativen Darstellungsform aus dem Graphen an. Abbildung 4.12 auf Seite 164 gibt einen Überblick über den Handlungsverlauf (hier definiert als Zuweisung von Charakteren zu Szenen, in denen sie sprechen) von drei ausgewählten Stücken, welcher sich visuell grob an U-Bahn-Stationsplänen orientiert. Die Visualisierungssoftware wurde aus den „Comic Book



(a) Verknüpfung der Abfolge von einzelnen Sprechakten, Schwellwert: 0



(b) Schwellwert: 1



(c) Schwellwert: 2



(d) Schwellwert: 3

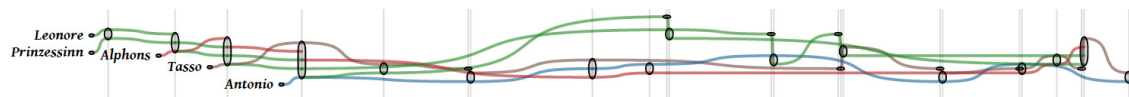
Abbildung 4.11: Einfluss von Interaktions-Schwellwerten für die Graphinduktion auf die sichtbare Struktur: Die Kante zwischen *Danton* und *Robespierre*, die in (b) und (c) noch zwei Gruppen des Stücks verbindet, besteht in (d) nicht mehr und der Graph zerfällt in viele Einzelkomponenten.

Narrative Charts³⁸ von Nancy Iskander, Matthew Thorne und Craig Kaplan übernommen.

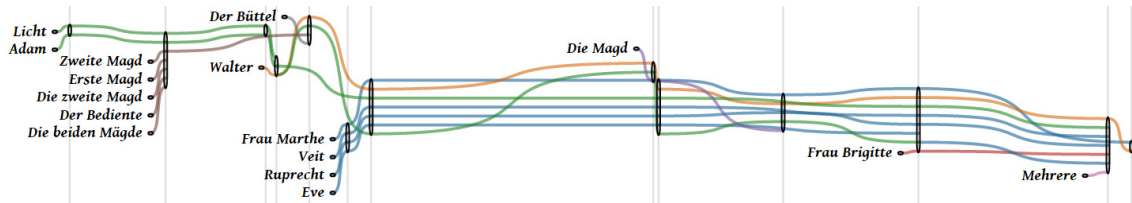
Für solche Aggregation bieten sich die im Dokument enthaltenen Struktureinheiten des Stücks an, eine andere, willkürlichere Einteilung, etwa in Abschnitte gleicher Wortanzahl oder gleicher Sprechaktanzahl, kann alternativ vorgenommen werden.

Werkzeuge, wie die vorgestellten Visualisierungen von Struktur und Dialogen, erlauben ein *Distant Reading* und zu einem gewissen Grad auch Vergleiche zwischen Stücken oder sogar Sammlungen von Stücken. Ein solcher, hin zu größeren Dokumentsammlungen skalierender Ansatz kann auch neue Perspektiven in der Fachwissenschaft eröffnen, da er die Betrachtung breiterer Kollektionen ihrer Forschungsgegenstände ermöglicht. In

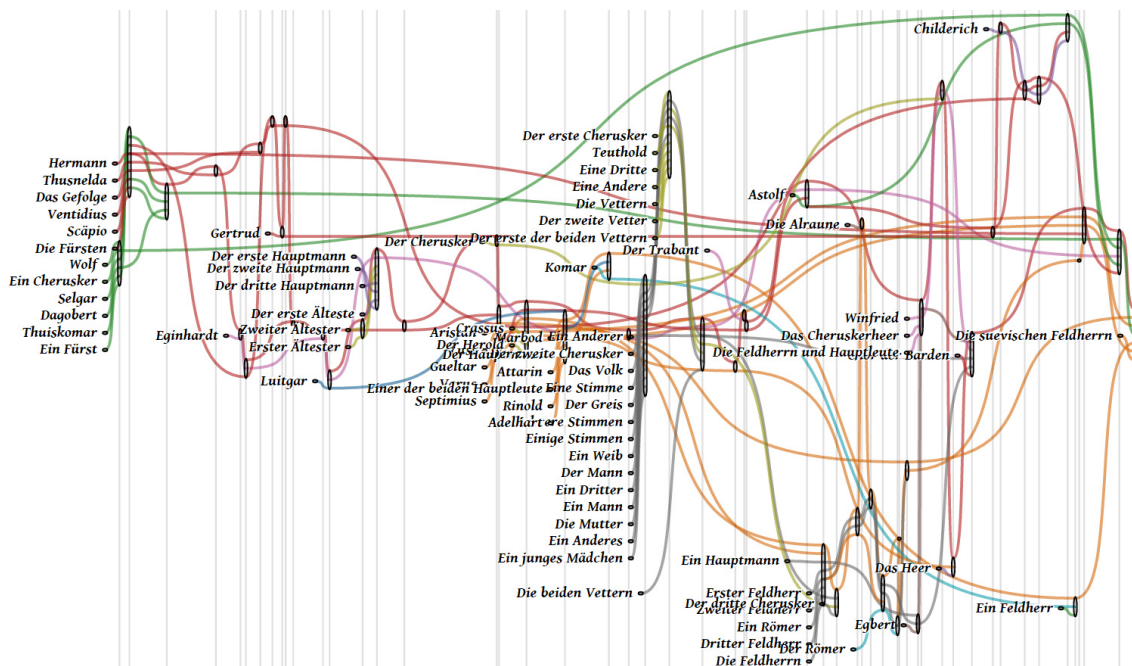
³⁸http://csclub.uwaterloo.ca/~n2iskand/?page_id=13



(a) Johann Wolfgang Goethe: *Torquato Tasso*



(b) Heinrich von Kleist: *Der zerbrochene Krug*



(c) Heinrich von Kleist: *Die Hermannsschlacht*

Abbildung 4.12: Beispiele für den Handlungsverlauf im **DRAMEN-Korpus**: Die Stücke mit wenigen Charakteren und Szenen, (a) und (b) lassen sich gut automatisch abbilden, während bei (c) zu viele Elemente die Übersichtlichkeit beeinträchtigen.

[Hir11] wurden alle seit dem Jahr 1950 erstellten kritischen Editionen englischsprachiger Dramen der Renaissance nach Dramatiker gruppiert gezählt. Mit 1285 Editionen entfallen mehr als die Hälfte der gelisteten Werke allein auf Shakespeare. Eine Stärke der e-Humanities ist es, den Fokus der Forschung potentiell weg von einem kleinen Kanon und hin zu einer Gesamtschau von Quellenmaterial zu bringen.

In der Praxis zeigen sich allerdings gerade in der Verfügbarkeit und im Erschließungsstadium größerer Korpora noch erhebliche Verbesserungsmöglichkeiten. Ein positives

Beispiel ist das [SHC-Korpus](#), wobei auch dort trotz validem [TEI](#)-Formats ein Import der Dramen nicht einfach ist: [TEI](#) erlaubt sehr viele verschiedene Verschachtelungsarten und alternative Modellierungen für gleiche Phänomene. Zudem sind gerade die älteren Stücke eher selten in Szenen eingeteilt (zuweilen nicht einmal in Akte), was nur eine werkswerte Auswertung ohne die Betrachtung solcher Segmentierungen erlaubt oder neue Mechanismen zur Detektion eines „Szeneriewechsels“, z. B. aus Regieanweisungen, erfordern würde.

Für eine fachwissenschaftliche Auswertung ist jedoch oft nicht nur die Makro-Skala von Bedeutung, sondern auch die Aufdeckung kleiner Nuancen und Varianzen in einzelnen Texten. In [Abbildung 4.13](#) werden die Buchübersetzungen aus dem [OTHELLO-Korpus](#) ausschnittsweise verglichen. Als Visualisierungswerkzeug wurde TraViz [[JGBS14](#)] verwendet. Dort wird ein graphbasiertes Verfahren zur Alignierung der Editionen eingesetzt, welches sich grundsätzlich auch direkt in Kadmos implementieren ließe. Auf diese Weise könnte z. B. auch die gemeinsame Konzeptzugehörigkeit von Begriffen zur Alignierung herangezogen werden, wodurch sich sogar ein Überschreiten von Sprachgrenzen realisieren ließe – entsprechende Erweiterungen des Systems stehen aber zur Zeit noch aus.

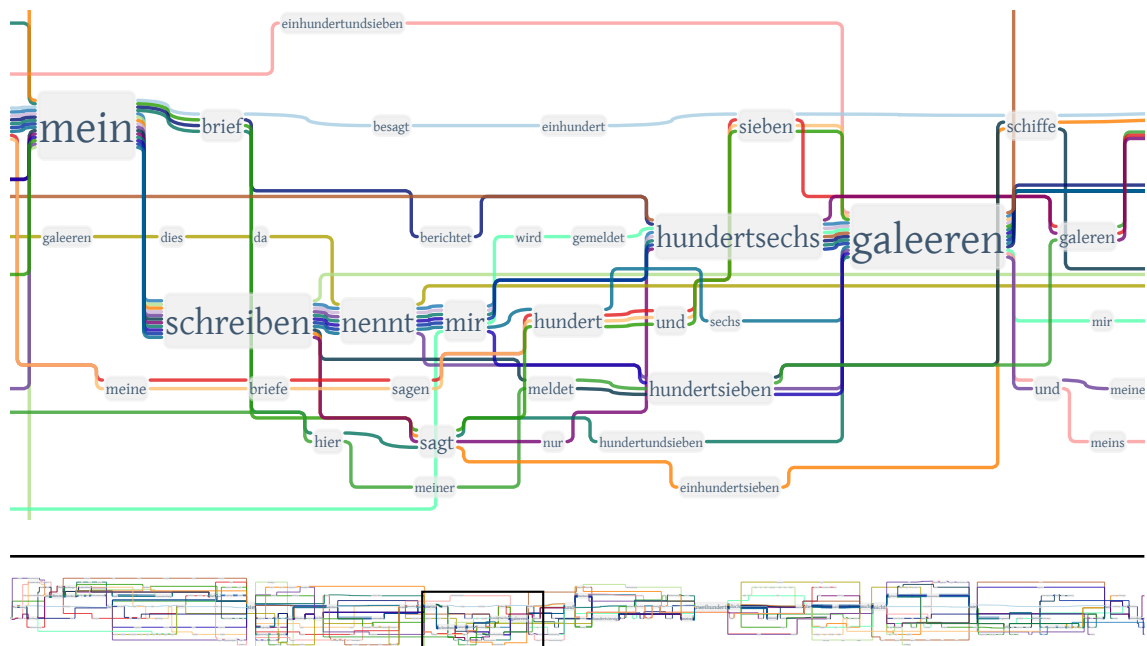


Abbildung 4.13: Übersetzungsvarianten im [OTHELLO-Korpus](#)

Das Anwendungsbeispiel des [OTHELLO-Korpus](#) wurde bereits in [[JEB15](#)] vorgestellt. Besonderes Augenmerk liegt auf den hier sichtbar werdenden syntagmatischen und paradigmatischen Beziehungen zwischen dem verwendeten Vokabular. Auffällig ist auch

die häufige Reduzierung von „*a hundred and seven galleys*“ auf 106 in den Übersetzungen. Eine naheliegende Erklärung ist die Vermeidung sperriger metrischer Formen, wie sie z. B. „*ein hundred und sieben*“ aufweist. Ein erweiterter Erklärungsversuch könnte auch die Beeinflussung späterer Übersetzer durch die früheren Editionen sein.

Diese Form des Textvergleichs ist natürlich nicht auf Dramen beschränkt, sondern kann für jede Art von Paralleltexten, teilalignierten Dokumenten, Zitationen oder anderem so genannten „*Text Re-Use*“ durchgeführt werden.

Die letzten Beispiele haben einen Eindruck davon vermittelt, welche Möglichkeiten sich für eine visuelle Unterstützung der Analyse von Dramen (als eine Form komplexer Texte) bieten. Im folgenden Abschnitt wird ein anderer Ansatz vorgestellt, diese Komplexität als Grundlage für Vergleiche über Dokumente hinweg zu nutzen.

4.4.2 Informationstheoretische Komplexitätsbetrachtungen

In diesem Abschnitt werden Beiträge zur Untersuchung der Komplexität von Sequenzen im Sinne der Informationstheorie beschrieben, wie sie Mitte des 20. Jahrhunderts maßgeblich von Claude Shannon etabliert wurde [Sha48]. Ein zentraler Begriff ist dabei die Entropie³⁹ als eine Möglichkeit zur Quantifizierung von Unsicherheit über künftige Nachrichten einer Quelle oder über Ereignisse, die von Zufallsgrößen abhängen.

Das im Folgenden (in aller gebotenen Kürze) vorgestellte Maß wurde gemeinsam mit Gerhard Heyer und Jürgen Jost im Rahmen der Arbeiten an [EHJ15] konzipiert. Ziel war die theoretisch fundierte Definition einer abstrakten numerischen Größe zur Abbildung der Komplexität innerhalb einer Sequenz von „Zeichen“, welche auch vorangehende zeichenübergreifende Einheiten zur Einschätzung der Vorhersagbarkeit eines nachfolgenden Zeichens berücksichtigt. Es steht beispielhaft für zahlreiche ähnliche Maße, die durch verwandte statistische und informationstheoretische Überlegungen erlangt werden können. Die thematische Breite der statistikbasierten Erfassung von Komplexität kann in dieser Arbeit leider nicht verdeutlicht werden. Es sei aber beispielhaft auf [Li90] verwiesen, wo eine in diesem Zusammenhang aufschlussreiche Abhandlung über *Mutual Information* und Korrelation in Sequenzen (u. a. auch natürlicher Sprache) gegeben wird.

³⁹Die informationstheoretische Entropie wird häufig mit der gleichnamigen Größe aus der Thermodynamik vermischt, was jedoch (zumindest ohne die Berücksichtigung neuerer Überlegungen zu Quanteneffekten) nicht gerechtfertigt ist, wie [JB72] zeigt.

Bei der Herleitung und Motivation des entwickelten Maßes wird der Einfachheit halber auf eine Unterscheidung zwischen der (eigentlich wahrscheinkeitsbasierten) Entropie H und einem auf Beobachtungsfrequenzen beruhenden Entropieschätzer $\hat{H}$ verzichtet, ebenso wie auf die Einführung eines daraus resultierenden Korrekturterms, etwa nach [Mil55].

Die Sequenzen, deren Komplexität durch die Maßzahl ausgedrückt wird, können beliebige Abfolgen sein, etwa von Ereignissen, Ergebnissen (diskreter) Messungen oder notierten Symbolen in einer Symbolkette. Als einheitliche mathematische Basis wird dafür der folgende abstrakte Formalismus verwendet:

Gegeben ein Lexikon L als nichtleere Menge von m Symbolen t , lassen sich eine Zeichenfolge A der Länge l und die daraus resultierenden n -Gramm-Sequenzen A_n der Längen $l - n + 1$ definieren:

$$L = \{ t_i : 1 \leq i \leq m \} : m \in \mathbb{N}^+ \quad (4.1)$$

$$A = \langle a_1, a_2, \dots, a_l \rangle : l \in \mathbb{N}^+, a_j \in L \quad (4.2)$$

$$\begin{aligned} A_2 &= \langle (a_1, a_2), (a_2, a_3), \dots, (a_{l-1}, a_l) \rangle \\ &\vdots \\ A_n &= \langle (a_1, a_2, \dots, a_n), (a_2, a_3, \dots, a_{n+1}), \dots, (a_{l-n+1}, a_{l-n+2}, \dots, a_l) \rangle : \\ &\quad n \in \mathbb{N}^+, n \leq l \end{aligned} \quad (4.3)$$

Die relative Häufigkeit (Frequenz) eines Zeichens t in A errechnet sich als:

$$f_A(t) = \frac{\sum_{k=1}^l \begin{cases} 1, & \text{wenn } a_k = t \\ 0, & \text{wenn } a_k \neq t \end{cases}}{l} \quad (4.4)$$

Die relative Häufigkeit einer Zeichenfolge S der Länge n in A_n ergibt sich analog:

$$f_{A_n}(S) = \frac{\sum_{k=1}^{l-n+1} \begin{cases} 1, & \text{wenn } a_{n_k} = S \\ 0, & \text{wenn } a_{n_k} \neq S \end{cases}}{l - n + 1} \quad (4.5)$$

Das Informationsmaß I eines Wahrscheinlichkeitswertes p , standardmäßig in Bit gemessen, folgendermaßen definiert:

$$I_b(p) = \begin{cases} 0, & \text{wenn } p = 0 \\ -p \cdot \log_b(p), & \text{wenn } p \neq 0 \end{cases} \quad (4.6)$$

$$I(p) = I_2(p) \quad (4.7)$$

Die Entropie H der Zeichensequenz A lässt sich unter Verwendung der relativen Auftrenshäufigkeiten zur Abschätzung von Auftretenswahrscheinlichkeiten nun wie folgt berechnen:

$$H(A) = \sum_{t \in L} f_A(t) I(f_A(t)) \quad (4.8)$$

Die gleiche Berechnungsvorschrift gilt für die Entropie der n -Gramm-Sequenz A_n :

$$H(A_n) = \sum_{S \in A_n} f_{A_n}(S) I(f_{A_n}(S)) \quad (4.9)$$

Unter der Blockentropie h_i der Zeichensequenz A mit Blocklänge i verstehen wir die Differenz:

$$\begin{aligned} h_1 &= H(A_2) - H(A) \\ h_i &= H(A_{i+1}) - H(A_i) \text{ für } i > 1 \end{aligned} \quad (4.10)$$

Ein einfaches Anwendungsbeispiel dieser Berechnungsvorschrift für zwei verschiedene Sequenztypen wird in [Abbildung 4.14 auf der nächsten Seite](#) gegeben. Dort wird für den Text aus dem [COPPERFIELD-Korpus](#) die n -gramm-Entropie berechnet, wobei er einmal als Wort- und einmal als Buchstabensequenz betrachtet wird. Die Blockentropien lassen sich jeweils als „Höhenunterschied“ auf der Y -Achse zwischen den einzelnen benachbarten Datenpunkten ablesen. Es zeigt sich, dass im Korpus Wort- n -gramme der Länge 5 und aufwärts nur noch so selten vorkommen (oder ganz fehlen), dass kein erkennbarer Entropie-Zugewinn durch Berücksichtigung der entsprechenden Blocklänge erreicht werden kann. Ebenso zeigt sich der gleiche Effekt für die Zeichensequenz – jedoch erwartungsgemäß bei deutlich größeren Blocklängen. Hierfür sind neben langen Eigennamen sicher auch fixe Phrasen im Text verantwortlich.

Ein „Zeichen“ im Sinne der hier vorgestellten Sequenzanalyse kann – wie gerade gezeigt

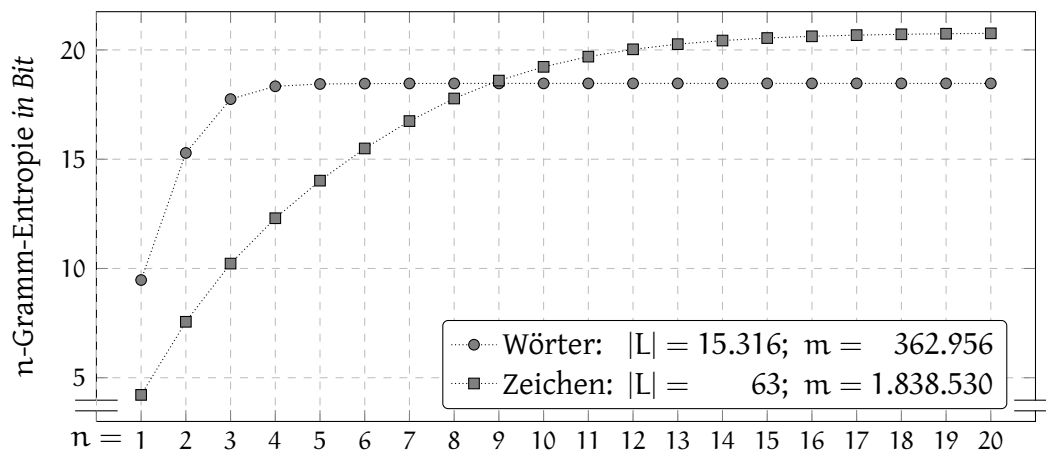


Abbildung 4.14: n-Gramm-Entropie für das **COPPERFIELD-Korpus** als Zeichensequenz und als Wortsequenz

– ein Schriftzeichen oder Wort⁴⁰ sein: Darüber hinaus kann es sich aber auch um ein **POS**-Tag, einen sprechenden oder auftretenden Charakter eines Bühnenstücks, das kategoriale Ergebnis einer Sprechakt-Klassifikation (Ansprache/Antwort, innerer Monolog, äußerer Monolog, ...) oder jeden beliebigen anderen diskretisierbaren Teil einer Abfolge handeln. Die Modellierung von Texten als Graphstruktur unterstützt die Extraktion verschiedener Abfolgen, etwa als Aggregation erreichbarer Annotationsknoten aus der Tokensequenz oder durch Umformung einer Sequenz von Strukturelementen. Zusätzlich können auch im Anschluss an die Graphenabfrage noch Transformationen durchgeführt werden, etwa bei Wortfolgen eine Silbenzerlegung oder die Reduktion auf das Vorhandensein von Interpunktion in den nicht normalisierten Types⁴¹.

Innerhalb eines Korpus kann dann dokumentübergreifend untersucht werden, ob die so ermittelte Komplexität der Dokumente eine interpretierbare Segmentierung erzeugt, die eventuell mit dokumentbezogenen Metadaten korreliert. Dadurch wird der Werkzeugkasten der algorithmisch gestützten Literaturanalyse (vgl. noch einmal [Joc13] oder [Ram11]) erweitert, welcher sich bisher eher aus einfacherer Statistik gespeist hatte, wie der Untersuchung von Satztlängen, dem Zählen des (abschnittsweisen) Personenumfangs oder der Kategorisierung von Monologlängen. All diese Auswertungsmethoden dienen der Verdichtung der dokumentinternen Kontexte auf analytisch motivierte Kennzahlen. Diese Kennzahlen können dann in die Gesamtinterpretation eines Werks oder einer

⁴⁰Hier in Form eines Types im Alphabet, das mit entsprechenden Token in der Sequenz korrespondiert

⁴¹In [DOK⁺16] z. B. werden der Interpunktion interessante statistische Eigenschaften zugesprochen: „A distinct role of the full stops in inducing the long-range correlations in texts is evidenced by the fact that [...] quantitative characteristics on the long-range correlations manifest themselves in variation of the full stops recurrence times along texts“

Werkssammlung eingehen. Je mehr Perspektiven dabei eingenommen werden können, umso mehr Anhaltspunkte ergeben sich für die Einordnung in den Gesamtkontext.

In Abbildung 4.15 wird die Summe aller Blockentropien ab h_2 (vorstellbar als „Höhen-differenz“ zwischen dem zweiten Datenpunkt und dem Plateau in Abbildung 4.14 auf der vorigen Seite) für deutsche Buchübersetzungen im OTHELLO-Korpus gegen ihr Erscheinungsjahr abgetragen⁴². Der sich abzeichnende zeitliche Verlauf steht nun für literaturwissenschaftliche Interpretationen zur Verfügung (inwieweit sich Übersetzer von vorherigen Editionen inspirieren ließen, inwieweit die Schriften dem literarischen Zeitgeist oder der allgemeinen Sprachentwicklung entsprechen, etc.). Neuere Editionen älterer Übersetzungen sind in der Abbildung durch das Symbol ➡ im Bezeichner (zwischen Originalübersetzer und Editor) gekennzeichnet.

Abbildung B/5 auf Seite 250 im Anhang erweitert diese Ansicht um den zusätzlichen Kontext englischer Ausgaben und deutscher Bühnenversionen. Es lässt sich erkennen, dass die vorgestellte Maßzahl in diesem Szenario ansatzweise zur Sprachsegmentierung geeignet ist. Grundsätzlich zeigt sich, dass damit vor allem relative Vergleiche innerhalb homogener Dokumentkollektionen sinnvoll sind, da viele Faktoren und Eigenheiten der Sprache Einfluss auf die Blockentropie nehmen können. Die Maßzahl kann auch in der automatischen Verarbeitung genutzt werden, wo sie z. B. zusätzliche Feature-Dimensionen für das Clustern von Dokumenten (nach verschiedenen ableitbaren Sequenz-Komplexitäten) bereitstellt.

Für Vergleiche, in denen Aussagen über mehrere stark unterschiedliche Werke hinweg getroffen werden sollen, sind derzeit noch Untersuchungen bezüglich der numerischen Normalisierung nach Alphabetgröße bzw. Sequenzlänge durchzuführen. Wie oben bereits angedeutet, ergibt sich durch die Nutzung von relativen Frequenzen zur Abschätzung von Wahrscheinlichkeiten eine systematische Fehleinschätzung. Diese ist umso signifikanter, je kleiner die Magnitude ist, die zwischen Typeanzahl und Tokenanzahl liegt. Ist grundsätzlich (wie bei Wortsequenzen natürlicher Sprache) die Gesamtzahl möglicher Types nicht bekannt, kann eine einfache Korrektur nicht vorgenommen werden. Ein Ansatz zur Behandlung dieses Problems ist die asymptotische Betrachtung der Varianz des Schätzers in Abhängigkeit des Type-Token-Verhältnisses, wie in [Har75] vorgestellt. Die grundsätzliche Aufgabe zur „Schärfung“ der Maßzahl im Kontext der e-Humanities liegt jedoch allgemein weniger in einer mathematischen Korrektur von Schätzungsfehlern,

⁴²damit soll der „Informationsgehalt“, der sich über weitere Distanzen als nur benachbarte Wörter erstreckt, abgebildet werden

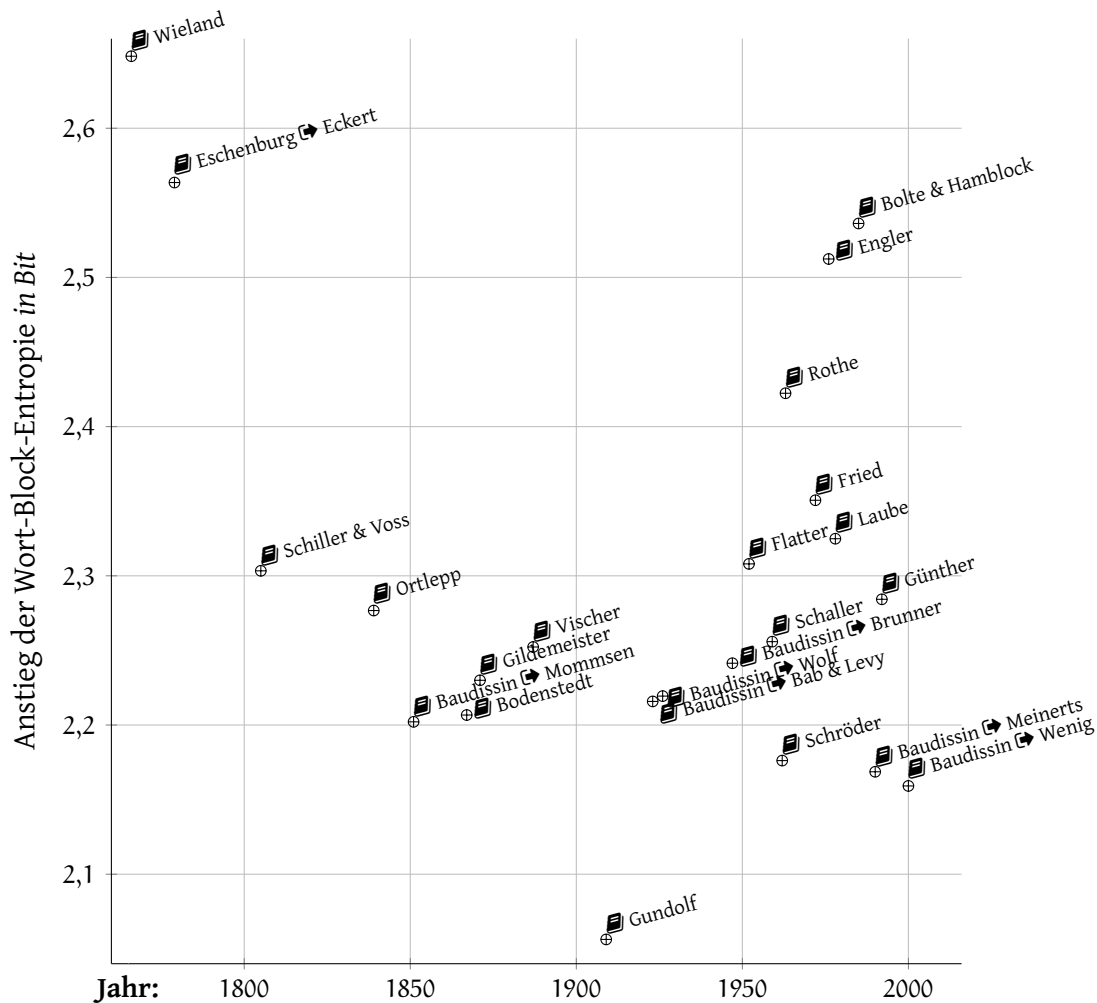


Abbildung 4.15: Blockentropie-basierte Maßzahl für deutsche Buchübersetzungen aus dem **OTHELLO-Korpus**

als vielmehr in ihrer umfassenden Dekorrelierung von der Sequenzlänge. Erst dadurch können empirische Vergleiche innerhalb von Dokumentkollektionen mit signifikant unterschiedlich langen Dokumenten vorgenommen werden. Daran geknüpft ist auch die notwendige Entwicklung von Methoden zur Abschätzung der Verlässlichkeit der Maßzahl in Abhängigkeit von der Sequenzlänge, mithin die Möglichkeit zur Unterscheidung von „Signal“ und „Rauschen“ im korpusweiten Vergleich.

Preliminäre Untersuchungen im Hinblick auf diese Normalisierungs-Problematik lassen noch keinen endgültigen Schluss über ein optimales Vorgehen zu. [Abbildung 4.16 auf der nächsten Seite](#) zeigt ein so genanntes QQ-Diagramm. Dieses kann verwendet werden, um einzuschätzen, ob zwei für alle Untersuchungsgegenstände gemessene Größen grundsätzlich eher den gleichen oder eher unterschiedlichen (Wahrscheinlichkeits-)Verteilungen

unterliegen. Zeigen die Datenpunkte im QQ-Diagramm einen stark linearen Zusammenhang, ist von zwei Verteilungen der gleichen Verteilungsart auszugehen – ein Fall, der einfachere Dekorellationsverfahren ermöglichen würde. Ein solcher Zusammenhang kann in den bisher untersuchten Fällen allerdings weder bejaht, noch verneint werden.

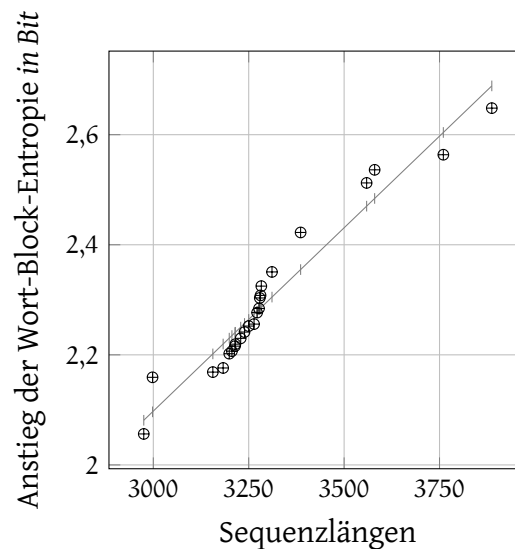


Abbildung 4.16: QQ-Diagramm der rangsortierten Sequenzlängen gegen die rangsortierte Maßzahl (ebenfalls [OTHELLO-Korpus](#)), mit linearer Regressionslinie

Trotz dieser offenen Fragestellungen bleibt festzustellen, dass die Informationstheorie viele neue Möglichkeiten bietet, Sequenzen auf einzelne, offenkundig bedeutungstragende Zahlenwerte zu reduzieren, die im Vergleich zwischen gleichartigen Dokumenten einen zusätzlichen Impuls für die Interpretation im *Distant Reading* liefern. Gleichzeitig kann gesagt werden, dass die Modellierung von Texten in einer Graphdatenbank alle benötigten Voraussetzungen für die schnelle Ableitung einfacher und komplexer Sequenzen in unübertroffener Klarheit und mit hoher Flexibilität schafft. Die Kombination beider technologischer Komponenten in einer leicht erweiterbaren Umgebung ermöglicht eine weitere experimentelle, datengestützte und theoriegeleitete Optimierung der Interpretierbarkeit der Maße, untersuchten Dokumente und Korpora. Hierin zeigt sich, dass nicht nur eine technische, sondern auch eine methodische Erweiterbarkeit gegeben ist.

4.5 Aufbau interner Indexstrukturen für lexikalische Ähnlichkeit

Der Zugriff auf Wörter, Personen und andere Objekte in der Graph-Datenbank über Zeichenketten ist ein wichtiger Aspekt der Interaktion mit den Daten. Kadmos konfiguriert Titan so, dass für diejenigen Properties, in denen Stringwerte gespeichert sind, ein externer Volltextindex verwendet wird. Über diesen ist es zum Beispiel möglich, eine trunkierte Suche im Stil von „Καδμ*“ zum Finden aller Einträge mit diesem Präfix durchzuführen. Ein externer Index eignet sich grundsätzlich auch für unscharfe Suchen, wie sie etwa bei Rechtschreibkorrektur-Vorgängen auftreten. Diese können bei der Textnormalisierung notwendig werden, etwa um Texterkennungs-Fehler automatisch zu beheben. Das Finden lexikalisch ähnlicher Wörter ist jedoch prinzipiell ein „lokales“ Problem, für das nicht unbedingt ein globaler externer Index aufgerufen werden muss.

Beispielhaft für eine Reihe solcher lokaler Index-Mechanismen soll hier demonstriert werden, wie die Editierdistanz (im Sinne der Levenshtein-Distanz ⁴³) mittels einfacher Konstrukte direkt in der Property-Graph-Datenbank umgesetzt werden kann.

Die Editierdistanz zweier Wörter ist die minimale Anzahl an einzeln und sequenziell ausgeführten Buchstabenlöschungen, -einfügungen und -ersetzungen, durch die sie ineinander überführt werden können. Für zwei gegebene Wörter kann sie sehr effizient mittels dynamischer Programmierung ermittelt werden. Für den Abgleich eines Wortes mit einem großen Lexikon ist eine auf paarweisen Vergleichen beruhende Suche nach ähnlichen Wörtern nicht praktikabel⁴⁴.

Zum schnellen Auffinden ähnlicher Wörter müsste eine „mutierte“ Umgebung für alle bekannten Wörter geschaffen werden. Das bedeutet konkret, dass eine Speicherung aller Realisierungen von (wiederholter) Einfügung, Löschung und Vertauschung mit Verweis auf das Originalwort durchzuführen wäre. Es ist leicht einzusehen, dass damit der benötigte Speicherplatz enorm groß ist, und durch die Einfügungsoperation in jedem Fall eine Begrenzung auf k Mutationsschritte nötig ist, um eine endliche Datenmenge zu erhalten.

⁴³nach bereits in [Lev66] skizzierten Überlegungen zu fehlertoleranten Binärcodes für die Datenübertragung

⁴⁴auch wenn sie sich über ein *Pruning* nach passenden Wortlängen und ggf. die Nutzung transitiver Eigenschaften der Editierdistanz in begrenztem Umfang beschleunigen lässt

Im FastSS-Algorithmus [BHS07] wird eine Umgebung nur durch Löschooperationen gebildet. Diese sind prinzipiell (in FastSS über ein *Dictionary*) rückverfolgbar und ergeben so eine Einfüge-Operation, über die in Richtung eines existierenden Wortes navigiert werden kann. Durch die Natur von Löschooperationen ergibt die so erzeugte Datenstruktur einen (nicht unbedingt zusammenhängenden) gerichteten azyklischen Graphen. Diese Graphenstruktur wird im Papier zu FastSS nicht explizit erwähnt, im Rahmen dieser Arbeit soll die Datenstruktur aber im Folgenden als (Property-)Graph beschrieben werden. Die Implementierung einer solchen Indexstruktur erfolgt direkt⁴⁵ in der Graphdatenbank über „Hilfsknoten“, ähnlich der in [RN11] als „represented [...] endogenous to the graph as an embedded tree“ beschriebenen Zugangsform. Abbildung 4.17 zeigt, wie die „Löschungsgraphen“ zweier ähnlicher Wörter verbunden sind.

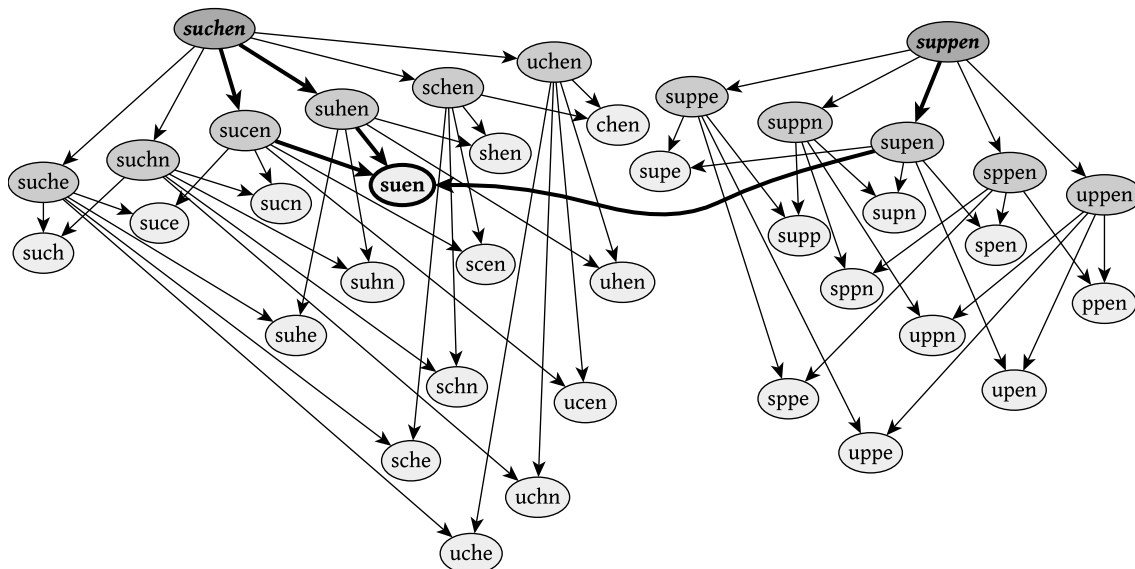


Abbildung 4.17: Die Löschungsgraphen der Tiefe 2 für *suchen* und *suppen* enthalten einen gemeinsamen Knoten

In den folgenden Beispielen wurden die eine Million häufigsten Wörter des deutschen Wortschatzes (entsprechend des Datensatzes *deu_words_1m* aus [QFH12]) mit ihren Umgebungen der Löschungstiefe $k = 3$ in Form eines Property-Graphen repräsentiert. Wörter, die aus mehr als 20 Buchstaben bestehen, wurden dabei nicht berücksichtigt. Die Verteilung der Wortlängen in der Wortliste ist in Abbildung 4.18 auf der nächsten Seite abgetragen.

Trotz der Beschränkung auf die Abbildung von Löschooperationen werden in der Daten-

⁴⁵und durch die persistente Speicherung jederzeit ohne zusätzliche Lade- oder Indizierungszeiten zugreifbar

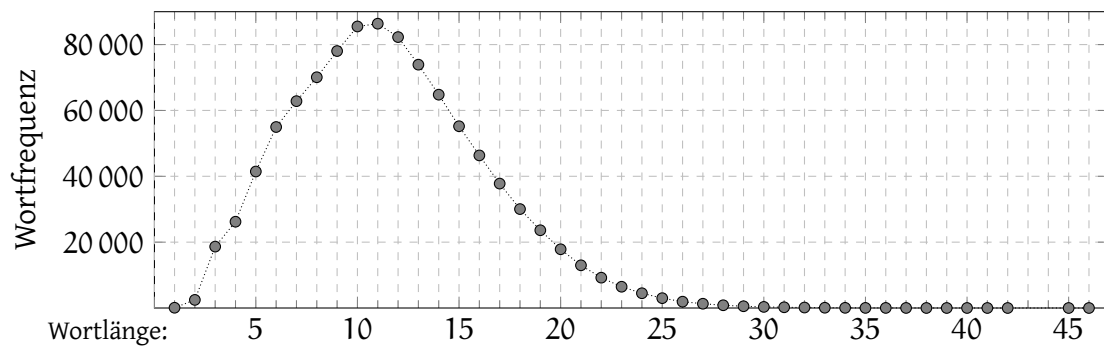


Abbildung 4.18: Wortlängenverteilung im untersuchten Wörterbuch

bank viele Hilfsknoten⁴⁶ erzeugt – je länger das Wort umso mehr Knoten ergeben sich. Die Anzahl der Knoten in der Umgebung lässt sich über einfache Abfragen ermitteln⁴⁷:

```

1 @graph.v(string:"bundesinnenminister").out.count          # => 18
2 @graph.v(string:"bundesinnenminister").out.out.uniq.count  # => 152
3 @graph.v(string:"bundesinnenminister").out.out.out.uniq.count # => 801

```

Das Wort `bundesinnenminister` enthält selbst nur 19 Zeichen. Insgesamt müssen jedoch 15726 Zeichen gespeichert werden. Dieses Wort teilt sich bei $k = 3$ keinen einzigen Hilfsknoten mit anderen Wörtern, z. B. `bundesfinanzminister` (Levenshtein-Abstand 4). Diese große und zudem „dünn“ mit ähnlichen Wörtern besetzte Umgebung für lange Wörter schlägt sich insgesamt negativ auf die Speichereffizienz nieder. Während die Wortliste als reiner Fließtext (*Plaintext*) nur ca. 14 MB groß ist, benötigt die Datenbank ganze 8310 MB Plattenplatz – ein Faktor im Bereich von 600.

Ist die Indexstruktur vollständig aufgebaut, kann sie zur Abfrage von Ähnlichkeiten genutzt werden. Abbildung 4.19 auf der nächsten Seite zeigt entsprechende Beispielpfade im Graphen. Dabei ist die Editierdistanz nicht in allen Fällen gleich der minimalen Pfadlänge zwischen zwei Wörtern, da eine einzelne Ersetzungsoperation für einen Buchstaben im Graphen über zwei Traversierungsschritte abgebildet wird. Jedoch kann dieser Umstand während des Abfragens kompensiert werden.

Über sogenannte *Side Effects* der Traversierungsrouten kann sich jeder *Traverser* die Stringwerte des letzten Ursprungsknotens einer Löschkante merken und bei der anschließenden

⁴⁶diese besitzen im Gegensatz zu Wortknoten keine weiteren Properties und lassen sich so von diesen einfach unterscheiden

⁴⁷Zur Effizienzsteigerung sollte die Zahl der Traversierungen noch durch den Aufruf von `.uniq` zwischen den einzelnen Traversierungsschritten begrenzt werden.

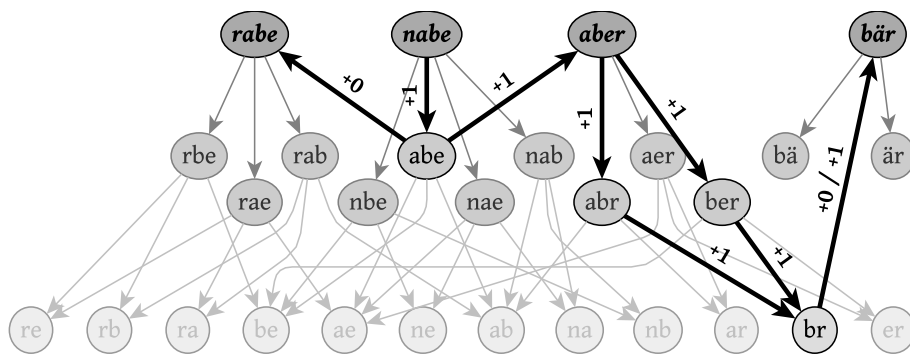


Abbildung 4.19: Distanzberechnungen bei der Traversierung des Löschrgraphen

den entgegengesetzten Traversierung mit dem neuen Stringwert vergleichen, um zu ermitteln, ob eine Ersetzungsoperation zu verzeichnen ist.⁴⁸

Im Beispiel kann von nabe aus rabe mit Abstand 1 erreicht werden (bei der „Aufwärts-traversierung“ wird die Ersetzungsoperation erkannt). Zu aber hingegen werden zwei vollständige Traversierungsschritte benötigt, die sich nicht zu einer Ersetzungsoperation kombinieren lassen, weshalb die Distanz 2 beträgt. Eine Traversierung von aber erreicht bär auf zwei verschiedenen Pfaden. Der mit den insgesamt geringeren Kosten berücksichtigt die mögliche Ersetzungsoperation und bestimmt eine Distanz von 2.

Die Abfrage aller im Wörterbuch enthaltenen Begriffe nach Levenshtein-Ähnlichkeit⁴⁹ zu einem gegebenen Suchwort kann nun ganz einfach durch Bildung aller Lösungsvarianten bis zur entsprechend gewünschten maximal zulässigen Lösungsanzahl⁵⁰ geschehen, wobei für diese und das Suchwort selbst eine Entsprechung im persistierten Löschrgraphen des Wörterbuchs gesucht wird. Von dort aus kann entsprechend des obigen Beispiels weiternavigiert werden. In diesem Vorgang sind viele Optimierungen möglich, etwa über den Abbruch von Traversierungen, die einen bereits vorher erreichten Knoten mit gleicher oder größerer bisher akkumulierter Editierdistanz erreichen. Um die Anzahl der frühen Abbrüche zu maximieren kann dann auch die Reihenfolge der abgefragten Lösungsvarianten optimiert werden, was jedoch nicht Gegenstand dieser Arbeit sein soll.

Über die hier vorgestellte, in die Datenbank integrierte Zugriffsstruktur für Stringähnlichkeit können Verfahren genutzt werden, in denen sowohl Dokumentdaten (wie beispielsweise Kookkurrenz-Statistiken) als auch lexikalische Ähnlichkeiten verwendet werden –

⁴⁸Alternativ kann eine Kantenproperty eingeführt werden, die zu jeder Löschung die Position speichert.

⁴⁹also geringer Levenshtein-Distanz

⁵⁰die kleiner oder gleich der bei der Initialisierung des Löschrgraphen verwendeten sein muss

so wie z. B. die automatische Taxonomieerzeugung nach [TKG15]. Generell ist natürlich anzumerken, dass für Probleme, in denen auch eine approximierte Ergebnismenge für gegebene Suchwörter und Editierdistanzen ausreichend ist, eine Vielzahl deutlich effizienterer Techniken (außerhalb von graphenförmigen Indexstrukturen) existiert, z. B. [MS04], wo mit Hilfsmitteln aus der Automatentheorie gearbeitet wird. Grundsätzlich ist die Option, solche zusätzlichen Verknüpfungspfade zwischen Elementen der Fachdomäne automatisch und ohne Technologiebrüche in der Datenbank nutzen zu können, ein Vorteil für die Flexibilität des Systems.

4.6 Das Graphenparadigma als interdisziplinäres Kommunikationsmittel

Die bisher vorgestellten Verfahren nutzen *Property Graphs* zur Abbildung zentraler Modellparameter der jeweiligen Anwendungsdomäne. Darauf aufbauende Arbeitsschritte, wie Filterung, Transformation und Auswertung, stützen sich zum großen Teil auf Abfragen der Graphdatenbank. Um Neuentwicklungen auf dieser Basis so zu gestalten, dass sie nicht als *Black Box* (wie in Abschnitt 2.1.2 auf Seite 21 beschrieben) wahrgenommen werden, ist es wichtig, bei den Nutzern des Systems ein Verständnis für seine genaue Funktionsweise herzustellen. Die in diesem Buch schon häufig angeklungene Diversität von Anwendungsfällen und -domänen sowie Nutzergruppen erschwert es, eine allgemeingültige Einführung mit geeigneten Analogien, Abstraktionen und Erklärungen zu geben.

Grundsätzlich sollte zudem nicht nur das einseitige Ziel verfolgt werden, ein Verständnis für die technischen Aspekte des Datenmodells bei den Nutzern zu fördern. Zusätzlich muss insbesondere auch ein umfangreicheres Verständnis der Domäne auf Seiten der Informatiker erreicht werden. Doerr beschreibt in [Doe03] die folgende Beobachtung als Arbeitshypothese für die Entwicklung des *CIDOC CRM*:

The problems computer scientists and system implementers have in comprehending the logic of cultural concepts seems to be equally as notorious as the inability of the cultural professionals to communicate these concepts to computer scientists.

Nur wenn diese Hürde überwunden ist, kann sichergestellt werden, dass geeignete Modellierungsansätze für die Abbildung der aus fachwissenschaftlicher Sicht wichtigen

Zusammenhänge in die Entwicklung des Werkzeugs einfließen können. Ein Lösungsansatz ist die iterative Entwicklung eines gemeinsamen Modells aus wohldefinierten und leicht verständlichen Bausteinen – wie sie im Fall von Property-Graphen durchaus gegeben sind.

Zu Beginn der Entwicklung (auf den Ebenen des Beschreibungsmodells und des Datenmodells nach [Fec16]) müssen technisch und inhaltlich passende Modelle gefunden und formalisiert werden. Für solche Aufgaben hat sich in der angewandten Informatik die Domänenanalyse als wohlverstandener Prozess etabliert. Dort wird üblicherweise mit etablierten Werkzeugen, etwa dem ERM gearbeitet, das sich (wie bereits beschrieben) auch für die Planung von Graph-Datenbanken eignet.

Das traditionelle Software-Engineering betrachtet die Herausarbeitung eines Domänenmodells oft als komplett vorgelagerten Schritt. In [Sch99] z. B. wird sie (in Analogie zu Beschreibungen der *Domain Analysis* aus dem englischsprachigen Raum) wie folgt charakterisiert:

[...] im Rahmen einer Domänenanalyse wird vor der Entwicklung von wiederzuverwendenden Komponenten eine allgemeine Analyse der betrachteten Problemdomäne vorgenommen, um Struktur- bzw. Verhaltensanalogien und -varianten innerhalb einer Domäne zu identifizieren.

Die Ziele der Domänenanalyse sind weiterhin sehr relevant. Seit der Jahrtausendwende haben sich jedoch im Software Engineering viele neue und agilere Methoden etabliert, die häufig auf vorgelagerte Domänenanalysen verzichten. Ohne eine dieser Methoden hier speziell herausheben oder bevorzugen zu wollen, bleibt festzustellen, dass der in dieser Arbeit präferierte prototypische und interdisziplinäre Entwicklungsansatz sowohl von frühzeitigen Modellierungsüberlegungen als auch von kurzen, agilen und iterativen Entwicklungszyklen profitiert.

Um die Flexibilität der Domänenanalyse zu erweitern, soll sie um die konsequente Interaktion mit realen Daten und Problemen ergänzt werden. Dies trägt auch der häufigen Beobachtung Rechnung, dass es sich in der Praxis sehr schwer gestaltet, ausschließlich durch abstrakte Betrachtungen die eigenen Denkmodelle und Arbeitsweisen zu externalisieren, zu kommunizieren und adäquat abzubilden.

Ein unter anderem auch in [SM99] verwendetes Zitat aus [MD85] fasst diesen Effekt gut zusammen:

We believe that experts cannot reliably give an account of their expertise: We have to exercise their expertise on real problems to extract and model their knowledge.

Wie schon besprochen, kann dieser Vorgang kein einseitiger Prozess des „Abfragens“ von Fachwissen sein. Stattdessen sollte eine gemeinsame Arbeit am Daten- und Domänenmodell angestrebt werden. Dafür ist es jedoch notwendig, dass die verwendeten Modellierungsmethoden und -konstrukte für alle Parteien transparent sind. Bezogen auf die hier beschriebenen Technologien ist ein Verständnis des Modells besonders wichtig, weil sich alle Analyseergebnisse, Filterungsvorgänge, Statistiken und Rankings auf das Datenbank-Schema und auf Datenbank-Abfragen zurückführen lassen.

Die in Apache TinkerPop integrierte Abfragesprache „Gremlin“ nutzt die Analogie einer „agentenbasierten“ Abfrage zur Verdeutlichung veränderlicher lokaler Abfragekontexte, Abfrage-Stacks und Aggregierungsmöglichkeiten. Dabei werden die *Traversers* des Abfrageprozessors, vgl. [Rod15], im Dokumentationsmaterial durch die für die Sprache namensgebenden Gremlins repräsentiert, etwa im angebotenen „30-Minuten-Tutorial“⁵¹.

Gremlins sind kleine agile Fabelwesen mit der Fähigkeit zur spontanen Replikation. Sie sind seit dem Erscheinen eines gleichnamigen Spielfilms Mitte der 1980er Jahre Teil der Populärkultur. Die *Apache Foundation* nutzt das Bild von sich koordiniert auf Graphstrukturen bewegend und aufsplittenden Gremlin-Horden zur Veranschaulichung der grundlegenden Funktionsweise des TinkerPop-Stacks und darauf ausgeführter Abfragen.

Auch in [RRS11] wird auf die Bedeutung von Metaphern hingewiesen, die wichtige Beiträge für die Kommunikation von Programminterna an die Nutzer einer Software leisten können. Sie erhöhen deren Fähigkeit zur qualifizierten Nutzung des Systems deutlich. Die Herausbildung eines passenden „Mentalen Modells“, welches der Nutzer von einem komplexen System oder Prozess aufbaut, kann durch die Nutzung von Analogien zu Bekanntem und durch passende Abstraktionen bei der Vermittlung der jeweiligen Grundeigenschaften und -funktionen unterstützt werden.

Die Nutzung der Gremlin-Metapher eignet sich für die grundsätzliche Beschreibung von Traversierungsmechanismen auf Property-Graph-Datenbanken, etwa von Vergleichen zur Ausführungsstrategie *breadth-first* im Vergleich zu *depth-first* oder von der Einbindung eines *Lookahead*, der Nutzung von Property-Filtern oder von Indexabfrage. Für

⁵¹<http://tinkerpop.apache.org/docs/3.2.0-incubating/tutorials/getting-started/>

jeden Schritt und jede lokale Umgebung der Abfrageausführung kann die Abstraktion auf verständlicher Ebene erfolgen: „Was tun die Gremlins in dieser Situation?“. Die Zahl „beteiligter“ Gremlins kann auch als Maßeinheit für die über die Eventsource-API mögliche quantitative Rückmeldung zum Bearbeitungsstand einer Abfrage verwendet werden und dadurch ein Gefühl für die mit der Anfrage verbundenen Rechenaufwände vermitteln.

Unter Nutzung dieser Herangehensweise für didaktische Zwecke und als generelle interaktive Testumgebung wurde „Gremlin's Property Graph Lab“ erstellt, dessen Oberfläche in Abbildung 4.20 zu sehen ist. Es handelt sich bei der Software um eine webbasierte Umgebung zur direkten Interaktion mit einem Property-Graph-System. Das *Graph Lab* ist dabei nicht auf Graphen nach dem Dokumenten-Schema von Kadmos beschränkt, sondern kann mit beliebigen Daten arbeiten, sobald diese im Property-Graph-Modell vorliegen. Damit ist es z. B. grundsätzlich vergleichbar mit „Gremlin-Bin“⁵², bietet aber deutlich tiefergehende Funktionen.

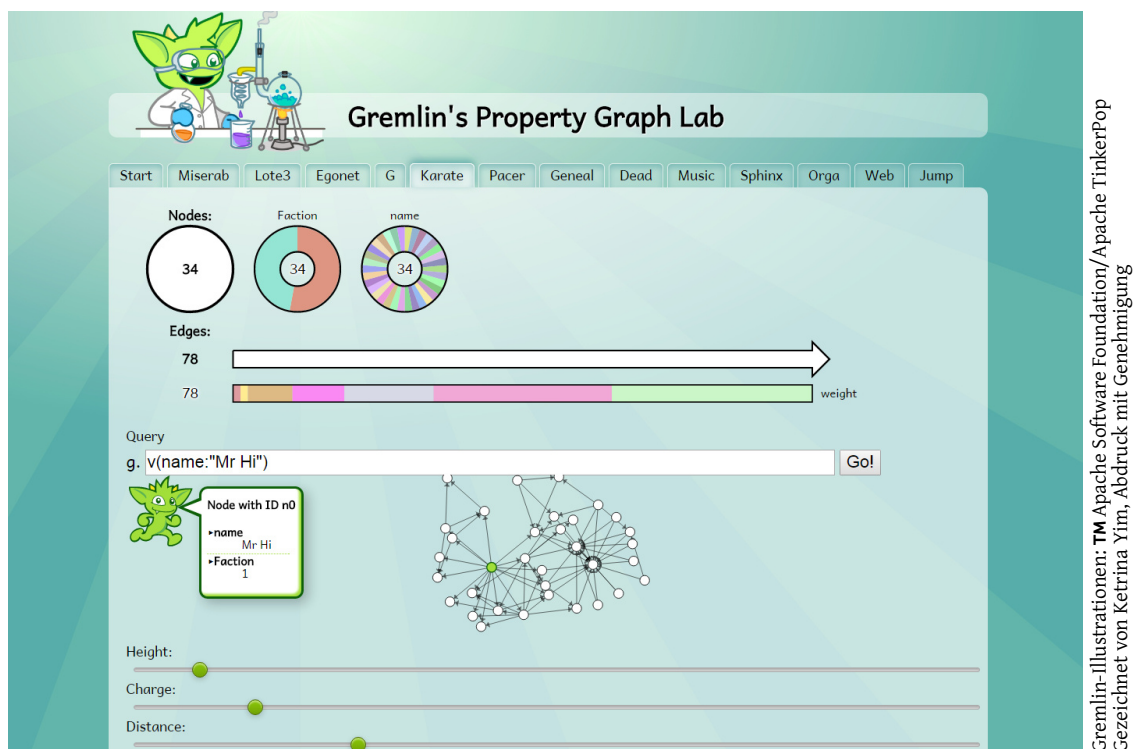


Abbildung 4.20: Oberfläche von „Gremlin's Property Graph Lab“ mit Beispielgraph aus den Daten von [Zac77]

Die Bündelung verschiedener Funktionen zur visuellen Analyse von und Interaktion mit den Graphdaten qualifiziert das *Graph Lab* als Werkzeug zur Vermittlung der Grundlagen

⁵²<http://gremlinbin.com/>

von Property-Graphen: Es ist dabei auf die Exploration kleiner übersichtlicher Beispielnetzwerke ausgelegt. Diese können allgemeiner Natur sein (Verlinkungsstrukturen, ein soziales Beziehungsnetzwerk, Musikstück-Interpreten-Netzwerke usw.) oder sich beim Schulungs-Einsatz in e-Humanities-Projekten speziell auf die Fachdomäne der Anwender beziehen. Dabei sind reale Daten (oder konsistente und exemplarische Ausschnitte davon) erfahrungsgemäß besser geeignet als künstliche Beispiele, wobei auch diese zum Erkenntnisgewinn beitragen können, wenn sie gemeinschaftlich erstellt werden.

Für das gewählte Beispielnetzwerk zeigt das System eine Graphenvisualisierung⁵³, ergänzt um einen visuellen Überblick über Knoten- und Kantenproperties und die Angabe ihrer Zusammensetzung, aggregiert nach Werten. Per Klick auf einen Wert wird eine entsprechende Abfrage erzeugt⁵⁴, in ein Eingabefeld eingetragen und ausgeführt. Die Abfrage kann auch jederzeit manuell verändert werden. Anfragen können dabei Ergebnisse in verschiedenen Antwort-Datentypen zurückliefern, wie in Abbildung 4.21 dargestellt ist.

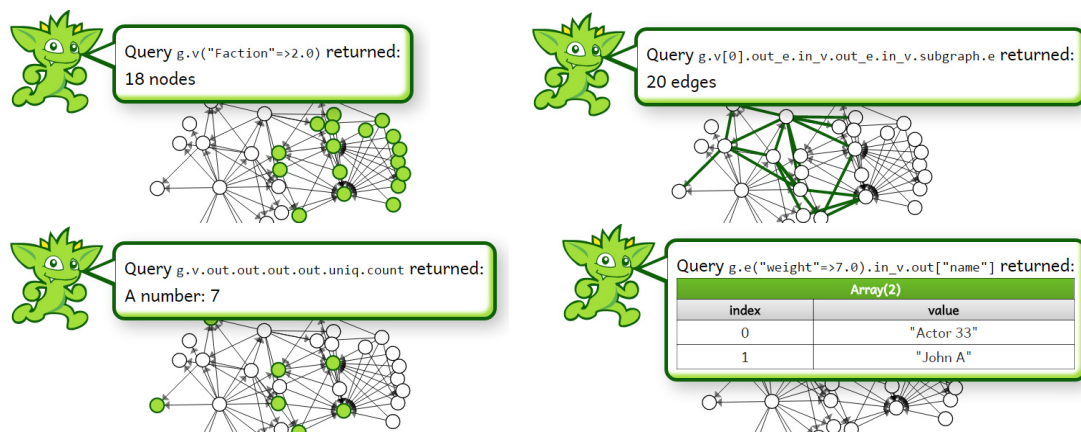


Abbildung 4.21: Abfrage- und Ergebnistypen in „Gremlin’s Property Graph Lab“

Besteht die Ergebnismenge aus Knoten oder Kanten, werden diese im Graphen hervorgehoben. Mit dem planarisierten Graphen kann auch interagiert werden: Eine Inspektion von Instanzdaten kann im Graphen durch Klicken auf das jeweilige Element erfolgen und durch Klicken und Ziehen mit der Maus ist es möglich, Knoten zu bewegen und so mit dem Layouting-Algorithmus zu interagieren.

⁵³ebenfalls über Sigmajs und [JVHB14]

⁵⁴Auch hier wird die in Kadmos genutzte Abfragesprache Pacer verwendet, die eng mit Gremlin verwandt ist, und verglichen mit dieser noch weniger Sonderzeichen benötigt. Zusätzlich besteht die Möglichkeit zur Mischung von Abfragerouten mit Ruby-Code.

Eine solche Umgebung vermittelt nicht nur allgemeine technische Grundlagen, sondern ermöglicht insbesondere ein schnelles Feedback für die ersten Schritte bei der prototypischen Entwicklung neuer Verfahren. Es gibt zudem Einblicke in die einzelnen relevanten Elemente sowie größeren Zusammenhänge der betrachteten Domäne. Damit lassen sich Verarbeitungsstrategien, zu erwartende Ergebnisse und damit verbundenes Fehlerpotential nicht nur benennen oder postulieren, sondern auch erkennen und visuell aufzeigen. In der Praxis hat sich gezeigt, dass auch komplexe Abfragen durch diese Form der Veranschaulichung kommunizierbar werden und damit gegenüber Fachwissenschaftlern alternative Berechnungsweisen qualifiziert erörtert oder diese gleich gemeinsam mit ihnen konzipiert werden können.

Property-Graphen und ihre Traversierungsmechanismen sind leicht verständlich. Die gemeinsame Entwicklung von Verfahren auf dieser Basis regt zum Mitdenken und Mitgestalten an und ersetzt vermeidbare *Black Boxes*.

Kapitel 5

Schlussbetrachtungen

After all is said and done there is usually more said than done.

US-Amerikanisches Sprichwort

5.1 Zusammenfassung

In den vorangegangenen Kapiteln wurde zunächst die Anwendungsdomäne der textbasierten e-Humanities als ein aktuelles und in vielerlei Hinsicht relevantes Forschungsfeld für die Informatik vorgestellt. Ein Überblick über bestehende Arbeitsweisen, Fragestellungen und Werkzeuge sowie ein Abriss über aktuelle Technologien zur Datenspeicherung und Abfrage haben verdeutlicht, dass die Nutzung von Graphdatenbanken zur Textrepräsentation in diesem Anwendungsgebiet zwar bisher nicht systematisch verfolgt wurde, aber dennoch ein nicht geringes wissenschaftliches wie technisches Potential birgt. Für das neuartige Recherchesystem Kadmos wurde auf dieser Grundlage ein für die Korpusrepräsentation geeignetes, für Text-Mining-Aufgaben gut einsetzbares und zudem leicht erweiterbares Datenmodell konzipiert. Auf diesem aufsetzend wurde eine skalierbare und flexible Systemarchitektur implementiert, mit der leichtgewichtige und reaktive Services in einem prototypzentrierten Entwicklungsprozess umgesetzt werden können. Mit diesem Basissystem wurden Verfahren für verschiedene Recherchevorgänge, wie eine facettierte und konzeptbasierte Suche und die explorative Erschließung von Vokabular entwickelt und vorgestellt. Ein letzter Schwerpunkt lag dann auf den Erweiterungsmöglichkeiten, die sich aus der Nutzung von Graphdatenbanken und dem Basissystem ergeben.

An dieser Stelle soll resümierend noch einmal ein Rückbezug auf die in Abschnitt [1.2 auf Seite 9](#) vorgestellten Forschungsfragen erfolgen:

Es konnte gezeigt werden, dass die kombinierte Verwaltung von Texten und Metadaten in einer bislang in der Literatur so nicht beschriebenen persistenten graphbasierten Repräsentationsform tatsächlich ein hohes Maß an Flexibilität für den textbezogenen Forschungsprozess bietet. Die Analyse diverser lokaler Kontexte interessanten Vokabulars wird ermöglicht, während Ergebnismengen von Anfragen beliebig nach ebensolchen Kontexten unter Berücksichtigung beliebiger Metadaten gefiltert und sortiert werden können. In Anfragen besteht zudem die Möglichkeit, beliebige neue „Verbindungen“ zwischen Datenelementen zu nutzen, die sich implizit aus der Traversierung des Graphen (oder aus externen Indexmechanismen) ableiten lassen. Nach einmaligen Aufwänden für den Import eines Korpus können zunächst Standardabfragen verwendet werden. Zusätzliche Aufwände entstehen erst wieder, wenn das System um forschungsfragenspezifische Funktionalität ergänzt werden soll. In diesen Fällen ist eine Kenntnis des internen Datenmodells und der Funktionsweise des Systems allerdings unumgänglich. Diese Arbeiten

sollten daher stets in gemischten Teams unter Berücksichtigung des aktuellen Stands der Forschung zur interdisziplinären Arbeitsweise, etwa nach [RS16], vorgenommen werden.

Es wurde weiterhin demonstriert, dass die Nutzung eines Graphdatenbanksystems eine geeignete Basis für ein skalierbares webbasiertes Textrecherchewerkzeug, welches auch mit großen Datenmengen umgehen kann, darstellt. Anfragen können parallel zueinander geschehen und der aktuelle Zustand des Systems wird direkt in den Abfrageergebnissen reflektiert. Zwischenergebnisse können unmittelbar und asynchron zurückgemeldet werden. Verschiedene interaktive webbasierte Recherche- und Visualisierungswerkzeuge wurden auf Grundlage dieser Technologie erstellt.

Anhand vieler Beispiele konnte gezeigt werden, dass sich durch die Modellierung der Forschungsdaten in Graphenstrukturen eine gute Ausgangslage für Verfahren sowohl quantitativer als auch qualitativer Natur bieten und dass sich eine Transition zwischen beiden Sichtweisen – mithin zwischen Aggregation und Detailansicht – problemlos realisieren lässt. Insbesondere für eine explorative Datenanalyse bieten sich vielfältige Möglichkeiten statistisch unterstützter Navigation in Textkollektionen. Da das Property-Graph-Modell grundsätzlich mit sehr einfachen Konstrukten beschrieben werden kann, gelingt es, darauf aufbauende Verfahren verständlich zu kommunizieren. Die Beschreibung der Forschungsprozesse auf der Ebene von Abfragearten und Traversierungsstrategien ermöglicht eine präzise Dokumentation (und kritische Betrachtung) des Forschungsprozesses.

Ferner konnte gezeigt werden, dass sich viele bekannte Methoden der automatischen Sprachverarbeitung auch in Systemen, die Graphdatenbanken zur Textrepräsentation nutzen, einsetzen lassen. Im Bereich des *Text-Mining* und des *Information Retrieval* ergeben sich viele neue Ansatzpunkte für alternative Verfahren. Wichtig ist hier noch einmal zu erwähnen, dass die Effizienz der so erstellten Verfahren wesentlich von ihrer „Lokalität“ abhängt. Vorberechnungsfreie Statistiken, die globale Informationen großer Korpora zusammenfassen, können nicht in interaktiven Anfrageszenarien eingesetzt werden. Daher dient die vorgestellte Technologie stets nur zur Ergänzung der bestehenden Werkzeugpalette, nicht als Ersatz für etablierte und meist stark optimierte traditioneller arbeitende Werkzeuge.

Schließlich wurde gezeigt, dass das Spektrum abbildbarer Untersuchungsgegenstände tatsächlich weit über reine Textdaten hinausreicht. Entsprechende Modellerweiterungen

wurden anhand praktischer Fragestellungen vorgestellt. So wurden u. a. verschiedene Zugriffsstrukturen zu Daten und Abfragemodi für Netzwerkanalysen realisiert. Anwendungsdomänen wie Philologie, Geschichte, Literatur- und Sozialwissenschaft wurden exemplarisch betrachtet. Durch die Nutzung eines vorhandenen Basissystems und einer mit vielen frei verfügbaren Werkzeugen ausgestatteten Basistechnologie gelingt es, die in erweiterten Anwendungsfällen entstehenden zusätzlichen Aufwände vergleichsweise gering zu halten.

Zur Kontrastierung dieser insgesamt sehr positiven Auswertung im Bezug auf die initialen Forschungsfragen soll nun noch einmal abschließend auf die bisher identifizierten Problempunkte eingegangen werden, die als bislang noch ungelöste Komplikationen oder als Hürden für die Adaption angesehen werden müssen.

5.2 Technologische und methodische Grenzen

Die größten technischen Limitationen ergeben sich aus der Natur echtzeit-abfragbarer Graphdatenbanken und der momentan noch vorhandenen Abgrenzung zu Graphprozessoren. Globale Analysen können in Graphdatenbanksystemen nicht effizient betrieben werden, wie z. B. in [CAH12] gezeigt wird, wo es dazu diplomatisch heißt: „[...] *the use cases with operations requiring local traversals in a large network are more suitable for the tested systems than those requiring traversals of the whole graph structure.*“.

Die einzelnen vorgestellten Verfahren enthalten noch keine „Mikro-Optimierungen“, wie *Caching*, Indexanpassungen, Datenkompression oder statistische Abschätzungsheuristiken. Deswegen sind die genauen Grenzen des Systems hinsichtlich der in „Echtzeit“ abfragbaren Korpusgrößen mit steigender „Globalität“ der Abfragen nicht abschätzbar.

Was sich jedoch schon jetzt sagen lässt, ist, dass die bisherigen, auf einzelnen Type-Token-Traversierungen beruhenden Möglichkeiten zur Volltextrückgabe großer Textstellen (wie ganzer Kapitel) zu langsam und ineffizient sind. Eine redundante Speicherung der größeren Textabschnitte direkt in den Strukturknoten erscheint (zumindest bei großen Korpora) nicht sinnvoll. Vorteilhafter wäre hier wohl die Nutzung eines dafür ausgelegten externen Systems, wie z. B. einer CTS-Instanz, welcher nur eine Referenzierung der gewünschten Textstelle per URN übergeben werden muss, um darüber effizient beliebig lange Abschnitte an Volltext auszulesen.

Die URNs können in Kadmos leicht aus der modellierten Struktur rekonstruiert werden. Bisher ist dies jedoch nur für „kanonische Texte“ sinnvoll umsetzbar. Ein Korpus aus Forentexten, unedierte Briefsammlungen oder Ähnlichem könnte so nicht ohne (teils erhebliche) Aufwände in ein geeignetes Referenzierungsschema gebracht werden. Als Alternative wäre auch die Rekonstruktion einzelner oder mehrere XPath-Ausdrücke aus der in Kadmos gespeicherten Struktur denkbar, mit deren Hilfe unter Nutzung einer XML-Datenbank die Volltextrückgabe geschehen könnte. Letztlich würde ggf. sogar eine einfache Hinterlegung der Texte in einer Verzeichnisstruktur als Plaintext-Dateien unterschiedlicher Granularität ausreichen. Fakt ist, dass für dieses Problem (auch mangels Anwendungsfall) bisher keine Lösung umgesetzt wurde.

Es ist allgemein wichtig, an dieser Stelle noch einmal darauf hinzuweisen, dass Kadmos keine umfangreiche Sammlung vorgefertigter Services anbieten möchte, wie sie etwa in der CLARIN-Infrastruktur vorgehalten werden. Die eigentliche Stärke liegt in der Möglichkeit, mit geringem Aufwand und hoher Flexibilität neue maßgeschneiderte Services auf Basis der zuvor in Graphen-Form erfassten Korpora in einem iterativen und idealerweise interdisziplinären Prozess erstellen zu können.

Auf der Basis des hier vorgestellten, hauptsächlich auf Backend-Aspekte ausgerichteten Systems wurden bereits erste Arbeiten Dritter durchgeführt, so z. B. [CJBS16]. Dabei haben sich bisher keine konzeptionellen Schwächen oder technischen Hürden gezeigt.

Eine methodische Grenze ergibt sich eventuell aus der Art der Formalisierung, die das System benötigt. Dazu argumentieren van Zundert und Kollegen in [ZAB⁺12], dass es neben der durch Digitalisierung erzwungenen Formalisierung noch viele weitere Formen gibt, die zum Teil schon lange in den Geistes- und Sozialwissenschaften verwendet werden. Sie stellen fest:

[...] formalisation can be supported by computation, if we recognise formalisation as an integral part of humanities practice and not as a feature driven only by computation.

Es muss sich somit erst noch in der Praxis zeigen, ob es mit Kadmos gelingt, die ganze Bandbreite möglicher Konzeptualisierungen, Abstraktionen, Modelle und Formalismen abzubilden, die aus den Fachwissenschaften heraus entwickelt werden. Eine interdisziplinäre und forschungsfragengeleitete Entwicklung kann von der explorativen Herangehensweise sicher profitieren. Der Abgleich mit bestehenden Theorien-Frameworks

wird jedoch nicht ohne Änderungen oder entsprechende Ergänzungen des Werkzeugs geschehen können.

Das hier für die Arbeit mit Kadmos beschriebene, hauptsächlich experimentelle Vorgehensmodell muss – auch im Sinne der e-Sciences – künftig noch um komfortablere Möglichkeiten zur Herstellung von *Reproducible Research*¹ ergänzt werden. Für diese und weitere auf einen einheitlichen und offenen Forschungsprozess ausgerichtete Erweiterungen muss stärker als bisher eine Absprache und technologische Kompatibilität mit den einschlägigen Infrastrukturinitiativen eingeplant werden.

Aus methodischer Sicht muss auch ganz allgemein der Umgang mit Graphen als Untersuchungsmittel oder -gegenstand in den e-Humanities weiter kritisch hinterfragt werden. Die Validität von Aussagen, die sich aus einer (manuellen oder algorithmisch unterstützten) Interpretation von graphbasierten Visualisierungen ableiten, kann nicht pauschal für jeden Anwendungsfall bestätigt werden. Nicht nur das visuelle Abbild, sondern auch die Datenbasis, der Entstehungsprozess, die Identität der repräsentierten Objekte und alle dabei getroffenen Annahmen müssen in die Betrachtung einbezogen werden. Wenn im Zuge einer Netzwerkanalyse dann z. B. Zentralitätsmaße eingesetzt werden, muss sichergestellt werden, dass diese auf die vorliegende Art von Daten und Netzwerkart überhaupt anwendbar sind. Insgesamt muss bei allen beteiligten Forschern eine ausreichende *Network Literacy* geschaffen werden, vgl. dazu die Broschüre [CY15].

Hier existiert noch weiterer Forschungsbedarf, insbesondere zu Theorien für die Kategorisierung und Interpretation von Graphen und Netzwerken sowie von Verfahren, die auf dieser Datengrundlage operieren. Für die Domäne sprach- und wortbezogener Netzwerke wurde in diesem Zusammenhang z. B. mit [Zwe16] bereits ein erster Versuch unternommen. Auf Seiten der Graphinduktion textbasierter Netzwerke wird z. B. in [MT04] als erster und wichtigster Schritt einer technischen Umsetzung aufgeführt: „*Identify text units that best define the task at hand, and add them as vertices in the graph.*“ Das Modell von Kadmos konzentriert sich hingegen auf kleinteilige generische Einheiten, aus denen sich später größere „*Objects of Interest*“ zusammensetzen lassen. Ob beide Herangehensweisen im Sinne einer nachgelagerten Netzwerkinterpretation äquivalent sind, ist noch zu klären.

¹vgl. <http://reproducibleresearch.net/>, wo der US-amerikanische Statistiker Dave Donoho wie folgt zitiert wird: „*An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.*“

Die hier kurz zusammengetragenen offenen Punkte sind nicht zuletzt auch Ansporn für weitere Entwicklungen. Ein Abriss über mögliche künftige Entwicklungen der in diesem Buch beschriebenen Technologie wird im nun folgenden, letzten Abschnitt gegeben.

5.3 Ausblick

Eine besonders wünschenswerte Perspektive für die hier vorgestellten Arbeiten ist die weitere Öffnung für angrenzende Fragestellungen und Datensammlungen. Durch das Herausstellen synergetischer Aspekte gelingt es eventuell, Beiträge zur methodischen Überbrückung einzelner Disziplinen zu leisten. In [Gör11] schreibt Görz:

Mit der Möglichkeit einer einheitlichen semantischen Modellierung von Daten verschiedener Fachgebiete eröffnen sich auch Perspektiven für die Bearbeitung hochkomplexer Fragestellungen, die ein einzelnes Fachgebiet nicht zu leisten vermag.

Entsprechende Erweiterungen in Kadmos müssen zunächst fachwissenschaftlich motiviert sein, so dass eine Anknüpfung an bestehende Semantische Modelle und die Umsetzung von *Best Practices* im Sinne der Anwender geschieht. Auf dieser Basis kann dann schrittweise eine Integration verschiedener solcher Erweiterungen in gemeinsames konzeptuelles (und mit praktischen Werkzeugen unterfüttertes) Modell geschehen. Eine breite Anschlussfähigkeit kann dabei auch durch Im- und Exportschnittstellen geschaffen werden. Die Anreicherung der JSON-basierten Schnittstelle um Linked-Data-Auszeichnungen ist bereits jetzt mittels der in [SLK⁺14] beschriebenen Technologie möglich.

Im Zuge dieser Erweiterungsstrategie sollten zuallererst Maßnahmen getroffen werden, die Nutzerbasis zu verbreitern. Dies kann durch die üblichen Wege der wissenschaftlichen Dissemination geschehen, sollte aber speziell auch Verbreitungswege der e-Humanities berücksichtigen, etwa die Eintragung von Kadmos in entsprechende Übersichtslisten, etwa die Werkzeugliste des Projekts TAPoR ² oder in das DiRT Directory³. Zum Besseren Verständnis der Nutzungsweisen und ggf. zur Aufdeckung impliziter Anforderungen kann eventuell eine Untersuchung von Nutzerverhalten anhand entsprechend gestalteter Nutzeroberflächen in einer öffentlichen Kadmos-Instanz stattfinden. Dadurch wäre die Basis für eine kritische Analyse des Nutzungsverhaltens, wie sie etwa in [Gre14] für das

²<http://tapor.ca/>

³<http://dirtdirectory.org/>

mittlerweile eingestellte Projekt „MONK – Metadata Opens New Knowledge“⁴ durchgeführt wurde, gelegt.

Ein längerfristiges Ziel ist eine intensive Verknüpfung von Korpusdaten mit nichttextuellen Ressourcen, besonders aus dem Bereich des Kulturellen Erbes, wie Museumsbeständen, Archivalien, Fotos, Karten. Dabei ist insbesondere die Verwendung von LOD-Ressourcen angedacht, von denen entweder passende Untermengen physisch in die Graphdatenbank integriert werden könnten, oder Kadmos um eine Möglichkeit zur föderierten Abfrage erweitert werden muss, damit die Verknüpfungsinformationen auch als Filter- und Aggregationskriterium genutzt werden können.

Mit immer größeren betrachteten Datenmengen und immer komplexeren semantischen Zusammenhängen, die abgebildet werden können, ist zu überlegen, ob eine Erweiterung des Kadmos-Systems um die Möglichkeit, lang laufende Anfragen in einem zurückgestellten (*deferred*) Verarbeitungsvorgang auszuführen, sinnvoll ist. Dies käme einer *Batch*-Verarbeitung nahe, aus deren Ergebnissen dann statische Webseiten, persistierte Datensammlungen (im Graphenmodell oder losgelöst davon) sowie Reporting-Dokumente erzeugt werden könnten. Hierbei ist speziell auch die Verwendung von Graphprozessoren vorstellbar, welche für OLAP-Funktionalität optimiert sind. Im Idealfall könnten diese über die selben Schnittstellen angesteuert werden und die selbe Abfragesprache nutzen, wie die eingesetzte OLTP-orientierte Graphdatenbank. Hierbei gilt es auch, die allgemeine technologische Entwicklung abzuwarten, und z. B. zu verfolgen, ob sich in diesem Umfeld Erweiterungen für Property-Graph-Modelle etablieren, die für Anwendungsfälle in Kadmos geeignet sind – etwa das in [JPT⁺16] vorgestellte Modell, das deklarative Operatoren zur Datenauswertung und die Verwaltung von sogenannten *Graph Collections* unterstützt.

Generell zeigt sich im Umfeld des *Graph-Processing* ein Trend zur Unterstützung von *Real-Time-Analytics* auf großen Datenbeständen, der sich insbesondere durch neue Entwicklungen im Bereich der *In-Memory*-Datenbanken umsetzen lässt. So können auch komplexere Algorithmen (z. B. fein abgestimmte Empfehlungssysteme) auf skalierbare Weise implementiert werden, vgl. z. B. [SJBL16]. Weitere Leistungsverbesserungen versprechen Systeme, die zur Datenverarbeitung Graphik-Prozessoren oder programmierbare Logikschaltkreise einsetzen, s. z. B. [FPT14] und [DCWY16]). Der größte Unterschied zwischen solchen Systemen und klassischen Graphdatenbanken ist die bislang noch fehlende Berücksichtigung der Persistenz der Daten, wobei für die beiden Aspekte der Datenhaltung

⁴<http://monkproject.org/> bzw. <http://monk.library.illinois.edu/>

und der Datenabfrage in Algorithmen eine Hybridlösung zweier Systeme denkbar ist. Generell wurden in den letzten Jahren auch erhebliche Fortschritte im verteilten Verarbeiten persistenter Graphstrukturen erreicht – vgl. etwa [SWX12] und [LGK⁺12] – und neue innovative Abfrage- und Aggregierungsmechanismen für solche Strukturen vorgeschlagen (s. z. B. [SSB16]), so dass auch auf der Ebene der verteilten Speicher- und Anfragesysteme auf großen Daten mittelfristig eine Eignung für Echtzeitabfragen zu erwarten ist.

Wenn in einem solchen Umfeld die effiziente und schnelle Berechnung globaler(er) Kennzahlen für Graphen möglich wird und dabei Graphdatenbanken und Graphprozessoren tatsächlich technologisch verschmelzen, scheint es lohnenswert, auch neuere Überlegungen aus der Graphentheorie in die Analyse von Korpora mit einzubeziehen, etwa wie in [Pit15] auf verschiedenen konzeptionellen Ebenen für die *Social-Media*-Analyse vorgeschlagen wird.

Voraussichtlich ab Januar 2017 werden systematische Untersuchungen und Weiterentwicklungen von Kadmos im Rahmen des Schwerpunktes „Digital Humanities“ am Leipziger Standort des *Competence Center for Scalable Data Services and Solutions*⁵ durchgeführt. Dabei wird der Fokus auf verschiedene Formen von Skalierung, die effiziente Nutzung verteilter Hardwarecluster und die Anwendbarkeit auf "Big Data" gelegt. Entsprechend sollen dort auch neue Technologien für Persistierung und Abfrage rezipiert werden.

Auf Basis der bereits umgesetzten sehr freien und feingliedrigen Textmodellierung könnte Kadmos schrittweise zum Werkzeug für das flexible Erstellen digitaler Editionen ausgebaut werden. Dafür müsste sicher zunächst eine exemplarische Umsetzung anhand eines konkreten Anwendungsfalls stattfinden, im Zuge welcher das System in dieser Hinsicht grundlegend ertüchtigt werden könnte. Anschließend muss eine verstärkte Berücksichtigung der Anforderungen und allgemeinen *Best Practices* aus der Editorik stattfinden. Dann könnten zahlreiche dokument-, versions- bzw. editionsübergreifende Analysen implementiert werden – beispielsweise für die Stemmatalogie, analog [KA16]. Grundsätzlich ist es auch reizvoll, in Kadmos Werkzeuge für die kompletten Arbeitsschritte zur Modellierung und Erstellung kritischer Editionen unter vollständiger gleichzeitiger Berücksichtigung mehrerer abweichender Textzeugen bereitzustellen, wie sie z. B. in [BR98] charakterisiert wird.

⁵<http://www.scads.de/>

Eine solch differenzierte Betrachtung von Varianten und Versionen von Textstellen muss nicht auf Editionen einzelner Werke beschränkt sein. Analyseverfahren zum Auffinden von *Text Re-Use* erlauben auch das Aufspüren von dokumentübergreifenden Parallelstellen und können genutzt werden, um sogenannte fragmentarische Texte abzubilden, die nur indirekt durch zitatweise Einbindung in andere Texte überliefert sind. Damit könnten bisherige Ansätze ergänzt werden, die vornehmlich CTS und TEI zur Repräsentation einzelner Fragmente nutzen, s. z. B. [BRBC09]. Ähnlich, wie eine Erweiterung für die Unterstützung moderner Formen von Hypertextualität würde dafür ggf. ein komplexeres Annotationsmodell wünschenswert, beispielsweise analog der in [LS08] (für die Textspeicherung in relationalen Datenbanken) beschriebenen Vorgehensweise. Mit steigender Komplexität des Datenbankschemas wäre in Kadmos dann eine selbstbeschreibende Schemadatei für eine Austauschbarkeit der auf der Festplatte gespeicherten Datenbankdateien zwischen Kadmos-Instanzen ohne zusätzliche Serialisierungsaufwände denkbar.

Im nächsten Schritt könnte ein System zur Versionskontrolle in die Graphdatenbank integriert werden. Dies kann entweder über Konstrukte der Modellierungssprache oder auf Systemebene geschehen. In jedem Fall müssten entsprechende Ansätze zur Verwaltung kontextualisierter Aussagen genutzt werden, ähnlich der in [Dam14] verwendeten Quadrupel als erweitertes Modell für annotierbare Tripel. Darüber wäre auch die Modellierung von Datenprovenienz als eine Kette aufeinander folgender Herkunftsangaben und Bearbeitungsschritte denkbar. Die grundsätzliche Eignung von Graphdatenbanken für solche Daten wurde bereits in [VMZ⁺10] nachgewiesen.

Perspektivisch ist zudem denkbar, Layoutinformationen von Dokumenten direkt in das Textdatenmodell einzubeziehen. Dadurch könnten weitergreifende Analysen in verschiedenen Anwendungsszenarien stattfinden: Bei Zeitungskorpora könnte etwa die relative Anordnung von Themen und der ihnen jeweils eingeräumte Platz untersucht werden, oder der relative Anteil von Werbung. In zweisprachigen Editionen könnten Paralleltexte so aligniert werden, wie sie im Druck zusammengebracht wurden. Die Layoutinformationen könnten eine Basis für die automatische semantische Auswertung und Transformation komplexer Druckerzeugnisse bilden, wie sie etwa Wörterbücher darstellen.

Weiterhin könnten Wort- und Annotationssequenzen mit Zeitstempeln ausgestattet werden, über welche Aussprachelänge und Pausen aus Korpora gesprochener Sprache abgebildet werden könnten – ggf. unter Berücksichtigung der Überlegungen aus dem (zwar grundsätzlich TEI-orientierten, aber auf Interoperabilität ausgelegten) Standard [ISO24624]. Damit ist eine Verknüpfung mit entsprechenden Ressourcen und Werkzeugen

möglich und Methoden der Automatischen Sprachverarbeitung könnten so um erste Informationen auf der Ebene von Multimodalität erweitert werden.

Auch der Anschluss an andere etablierte Werkzeuge sollte weiter ausgebaut werden, etwa in Form einer Abfragemöglichkeit mittels [CQL](#) für Anwendungen im linguistischen Bereich. In [\[BFW16\]](#) wird eine „Corpus Query Lingua Franca“ vorgestellt, die bisherige Sprachansätze systematisiert. Diese Erkenntnisse könnten mit dem graphbasierten Modell abgeglichen und für die Erstellung einer entsprechenden Kadmos-API genutzt werden.

Unter anderem durch solche grundlegend neuen Einsatzfelder und Funktionen werden zahlreiche neue Schnittstellen und darauf aufsetzende interaktive Visualisierungswerkzeuge benötigt. Eine verstärkte Verzahnung von Kadmos mit Techniken und Werkzeugen aus dem Bereich der *Visual Analytics* ist in diesem Zusammenhang sehr wünschenswert.



Über diese einzelnstehenden Zukunftsprognosen, Desiderata und offenen Arbeitsfelder hinausgehend ist es schwer, die Entwicklung der vorgestellten Technologien als Ganzes – mitsamt ihren Wechselwirkungen untereinander – vorherzuahnen. Noch schwerer fällt es, Prognosen über die Entwicklung ihrer Anwendungsdomäne abzugeben, zumal hier nur die Perspektive der Informatik eingenommen werden kann. McCarty verzichtet in [\[McC13b\]](#) auf eine einfache Antwort auf die Frage nach der Zukunft dieses breiten und veränderlichen, hier als e-Humanities vorgestellten Gebiets und seinem Einfluss auf die Geisteswissenschaften insgesamt: „[The response] could not be what we almost always get: a projection of current technical know-how into an imagined future“.

Stattdessen blickt er zurück und setzt Schlaglichter auf verschiedene Entwicklungen in der Wissenschafts-, Technik- und Sozialgeschichte und schließt mit den folgenden Worten, mit denen auch dieses Buch enden soll:

"What's driving the change, where is it heading and what might the humanities look like as a result?" Frighteningly, thrillingly, it's up to us.

Literaturverzeichnis

+ Gemeinsame Liste von Onlinequellen und konventioneller Literatur

- [AB14] ARTHUR, Paul L. (Hrsg.) ; BODE, Katherine (Hrsg.): *Advancing Digital Humanities – Research, Methods, Theories*. Palgrave Macmillan, 2014 (s. Seite 17)
- [ABG16] ALLINGTON, Daniel ; BROUILLETTE, Sarah ; GOLUMBIA, David: Neoliberal Tools (and Archives): A Political History of Digital Humanities. In: *Los Angeles Review of Books* (2016), (digital). <http://lareviewofbooks.org/article/neoliberal-tools-archives-political-history-digital-humanities> (s. Seite 18)
- [AH08] ALLEMANG, Dean ; HENDLER, James: *Semantic Web for the Working Ontologist – Effective Modeling in RDFS and OWL*. Morgan Kaufmann, 2008 (s. Seite 59)
- [AMWZ09] ARMSTRONG, Timothy G. ; MOFFAT, Alistair ; WEBBER, William ; ZOBEL, Justin: Improvements That Don’t Add Up: Ad-Hoc Retrieval Results Since 1998. In: *Proceedings of the 18th ACM international conference on Information & Knowledge Management (CIKM)* (2009) (s. Seite 121)
- [Ang12] ANGLES, Renzo: A Comparison of Current Graph Database Models. In: *Proceedings of the 28th International Conference on Data Engineering Workshops (IEEE ICDEW)*, 2012, S. 171–177 (s. Seite 56)
- [Ash56] ASHBY, William R.: *An introduction to cybernetics*. John Wiley & Sons, 1956 (s. Seite 21)
- [Aud08] AUDENAERT, Neal: Patterns of Analysis: Supporting Exploratory Analysis and Understanding of Visually Complex Documents. In: *Bulletin of IEEE Technical Committee on Digital Libraries (TCDL)* 4 (2008), Nr. 2, S. 1–14 (s. Seite 43)

- [Bab11] BABEU, Alison: *"Rome Wasn't Digitized in a Day": Building a Cyberinfrastructure for Digital Classics*. Council on Library and Information Resources, 2011
(s. Seite 27)
- [Bär16] BÄR, Manuel: *Graphetteer – A conceptual model for a graph driven gazetteer*. <http://www.geonet.ch/graphetteer/>. Stand: Jan 2016. Seminararbeit, MSc GIScience, University of Zürich
(s. Seite 141)
- [Bar02] BARABÁSI, Albert-László: *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Basic Books, 2002
(s. Seiten 8 und 50)
- [BBPI16] BARKER, Elton (Hrsg.); BOUZAROVSKI, Stefan (Hrsg.); PELLING, Christopher (Hrsg.); ISAKSEN, Leif (Hrsg.): *New Worlds from Old Texts: Revisiting Ancient Space and Place*. Oxford University Press, 2016
(s. Seite 141)
- [BDL⁺12] BURDICK, Anne; DRUCKER, Johanna; LUNENFELD, Peter; PRESNER, Todd; SCHNAPP, Jeffrey: *Digital Humanities*. The MIT Press, 2012
(s. Seite 17)
- [BDM⁺17] BEECHAM, Roger; DYKES, Jason; MEULEMANS, Wouter; SLINGSBY, Aidan; TURKAY, Cagatay; WOOD, Jo: Map LineUps: effects of spatial structure on graphical inference. In: *IEEE Transactions on Visualization and Computer Graphics (IEEE VIS 2016 Special Issue)* (2017), Nr. (im Druck). <http://dx.doi.org/10.1109/TVCG.2016.2598862>. DOI 10.1109/TVCG.2016.2598862
(s. Seite 154)
- [BDPS16] BOREK, Luise; DOMBROWSKI, Quinn; PERKINS, Jody; SCHÖCH, Christof: TaDiRAH: a Case Study in Pragmatic Classification. In: *Digital Humanities Quarterly (DHQ)* 10 (2016), Nr. 1, S. #3
(s. Seite 22)
- [BEK⁺04] BOX, Don; EHNEBUSKE, David; KAKIVAYA, Gopal; LAYMAN, Andrew; MENDELSON, Noah; NIELSEN, Henrik F.; THATTE, Satish; WINER, Dave: *Simple Object Access Protocol (SOAP) 1.1* / World Wide Web Consortium. Version: May 2004. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>. 2004 (Recommendation)
(s. Seite 235)
- [BFW16] BAŃSKI, Piotr; FRICK, Elena; WITT, Andreas: Corpus Query Lingua Franca (CQLF). In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, European Language Resources Association (ELRA), 2016, S. 2804–2809
(s. Seite 194)

- [BGH15] BÖHME, Christoph ; GEIPEL, Markus M. ; HANNEMANN, Jan: Metamorph: A Transformation Language for Semi-structured Data. In: *The Magazine of Digital Library Research (D-Lib)* 21 (2015), Nr. 5/6, S. #6 (s. Seite 153)
- [BHJ09] BASTIAN, Mathieu ; HEYMANN, Sebastien ; JACOMY, Mathieu: Gephi: An Open Source Software for Exploring and Manipulating Networks. In: *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, 2009, S. 361–362 (s. Seite 24)
- [BHS07] BOCEK, Thoma ; HUNT, Ela ; STILLER, Burkhard: *Fast Similarity Search in Large Dictionaries* / Department of Informatics, University of Zurich. 2007 (ifi-2007.02) (s. Seite 174)
- [Bie03] BIEMANN, Christian: Extraktion semantischer Relationen aus natürlich-sprachlichem Text mit Hilfe von maschinellern Lernen. In: SEEWALD-HEEG, Uta (Hrsg.): *Sprachtechnologie für die multilinguale Kommunikation, Proceedings der Frühjahrstagung der Gesellschaft für Linguistische Datenverarbeitung (GLDV)*, Gardez!-Verlag, 2003, S. 12–25 (s. Seite 137)
- [Bie06] BIEMANN, Chris: Chinese Whispers – an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In: *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing (TextGraphs-1)*, 2006, S. 73–80 (s. Seite 158)
- [BL12] BLANCO, Roi ; LIOMA, Christina: Graph-based term weighting for information retrieval. In: *Information Retrieval* 15 (2012), Feb, Nr. 1, S. 54–92 (s. Seite 108)
- [Bla10] BLAIR, Ann M.: *Too Much to Know – Managing Scholarly Information before the Modern Age*. Yale University Press, 2010 (s. Seite 6)
- [BLHL01] BERNERS-LEE, Tim ; HENDLER, James ; LASSILA, Ora: The Semantic Web. In: *Scientific American* 284 (2001), Nr. 5, 34–43. <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>. ISSN 0036–8733 (s. Seite 58)
- [BLK⁺09] BIZER, Christian ; LEHMANN, Jens ; KOBILAROV, Georgi ; AUER, Sören ; BECKER, Christian ; CYGANIAK, Richard ; HELLMANN, Sebastian: DBpedia - A crystallization point for the Web of Data. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 7 (2009), Nr. 3, S. 154 – 165. <http://dx.doi.org/10.1016/j.websem.2009.07.002>. DOI 10.1016/j.websem.2009.07.002. ISSN 1570–8268. The Web of Data (s. Seite 60)

- [BLLW10] BARCELÓ, Pablo ; LIBKIN, Leonid ; LIN, Anthony W. ; WOOD, Peter T.: Expressive Languages for Path Queries over Graph-Structured Data. In: *Proceedings of the 29th ACM Symposium on Principles of Database Systems (PODS)*, 2010, S. 3–14
(s. Seite 62)
- [BLW79] BIGGS, Norman L. ; LLOYD, E. K. ; WILSON, Robin J.: *Graph Theory 1736-1936*. Oxford University Press, 1979
(s. Seite 52)
- [BM14] BACHMANN-MEDICK, Doris: *Rowohlts Enzyklopädie*. Bd. 5. Auflage: *Cultural Turns: Neuorientierungen in den Kulturwissenschaften*. Rowohlt Taschenbuch Verlag, 2014
(s. Seite 16)
- [BN12] BOHNET, Bernd ; NIVRE, Joakim: A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) 2012*, 2012, S. 1455–1465
(s. Seite 126)
- [BNJ03] BLEI, David M. ; NG, Andrew Y. ; JORDAN, Michael I.: Latent Dirichlet Allocation. In: *Journal of Machine Learning Research* 3 (2003), S. 993–1022
(s. Seite 41)
- [BOH11] BOSTOCK, Michael ; OGIEVETSKY, Vadim ; HEER, Jeffrey: $\mathbb{D}^3$: Data-Driven Documents. In: *IEEE Transactions on Visualization and Computer Graphics* 17 (2011), Nr. 12, S. 2301–2309
(s. Seiten 142 und 155)
- [Bou13] BOUDIN, Florian: A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. In: *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2013, S. 834–838
(s. Seite 108)
- [BPD11] BOLLMANN, Marcel ; PETRAN, Florian ; DIPPER, Stefanie: Rule-Based Normalization of Historical Texts. In: *Proceedings of the Recent Advances in Natural Language Processing (RANLP) Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, 2011, S. 34–42
(s. Seite 94)
- [BPSM⁺06] BRAY, Tim ; PAOLI, Jean ; SPERBERG-MCQUEEN, C. M. ; MALER, Eve ; YERGEAU, François ; COWAN, John: *Extensible Markup Language (XML) 1.1 / W3C*. Version: August 2006. <http://www.w3c.org/TR/xml11>. 2006 (Recommendation)
(s. Seite 237)

- [BQHR12] BURKELL, Jacquelyn ; QUAN-HAASE, Anabel ; RUBIN, Victoria L.: Promoting Serendipity Online: Recommendations for Tool Design. In: *Proceedings of the iSchools iConference*, 2012, S. 525–526 (s. Seite 24)
- [BR98] BUZZETTI, Dino ; REHBEIN, Malte: Text Fluidity and Digital Editions. In: *Proceedings of the International Workshop on Text variety in the witnesses of medieval texts*, 1998, S. 14–39 (s. Seite 192)
- [Bra05] BRADLEY, John: Documents and Data: Modelling Materials for Humanities Research in XML and Relational Databases. In: *Literary and Linguistic Computing (LLC)* 20 (2005), Nr. 1, S. 133–151 (s. Seite 38)
- [BRBC09] BERTI, Monica ; ROMANELLO, Matteo ; BABEU, Alison ; CRANE, Gregory: Collecting Fragmentary Authors in a Digital library. In: *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2009, S. 259–262 (s. Seite 193)
- [BS12] BANK, Mathias ; SCHIERLE, Martin: A Survey of Text Mining Architectures and the UIMA Standard. In: CHAIR), Nicoletta Calzolari (. (Hrsg.) ; CHOUKRI, Khalid (Hrsg.) ; DECLERCK, Thierry (Hrsg.) ; DOĞAN, Mehmet U. (Hrsg.) ; MAEGAARD, Bente (Hrsg.) ; MARIANI, Joseph (Hrsg.) ; ODIJK, Jan (Hrsg.) ; PIPERIDIS, Stelios (Hrsg.): *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, European Language Resources Association (ELRA), 2012, S. 3479–3486 (s. Seite 136)
- [Bue12] BUERLI, Mike: *The current state of graph databases* / Department of Computer Science, Cal Poly San Luis Obispo. 2012 (Report) (s. Seiten 55 und 56)
- [Bur14] BURNARD, Lou: *What Is the Text Encoding Initiative? – How to Add Intelligent Markup to Digital Resources*. OpenEdition Press, 2014 (Encyclopédie Numérique) (s. Seite 37)
- [Bus45] BUSH, Vannevar: As We May Think. In: *The Atlantic* (1945), Jul, S. 1–8 (s. Seiten 2 und 109)
- [Bus80] BUSA, Roberto: The Annals Of Humanities Computing: The Index Thomisticus. In: *Computers and the Humanities* (1980), Nr. 14, S. 83–90 (s. Seite 15)
- [Bus04] BUSA, Roberto: Perspectives on the Digital Humanities. In: *A Companion to Digital Humanities*. Blackwell Publishing, 2004, S. Foreword (s. Seite 15)

- [Bus12] BUSSE, Dietrich: *Frame-Semantik: Ein Kompendium*. De Gruyter, 2012
(s. Seite 103)
- [Buz02] BUZZETTI, Dino: Digital Representation and the Text Model. In: *Reconsiderations of Literary Theory, Literary History* Bd. 33. 2002 New Literary History, S. 61–88
(s. Seite 38)
- [BW95] BARR, Michael ; WELLS, Charles: *Category Theory for Computing Science*. 2. Hertfordshire, UK, UK : Prentice Hall International (UK) Ltd., 1995. ISBN 0–13–323809–1
(s. Seite 54)
- [BW09] BURGHARDT, Manuel ; WOLFF, Christian: Stand off-Annotation für Textdokumente: Vom Konzept zur Implementierung (zur Standardisierung?). Version: 2009. <http://epub.uni-regensburg.de/14223/>. In: CHIARCOS, Christian (Hrsg.) ; CASTILLO, Richard E. (Hrsg.) ; STEDE, Manfred (Hrsg.): *Von der Form zur Bedeutung: Texte automatisch verarbeiten / From form to meaning: processing texts automatically: Proceedings of the Biennial GSCL Conference 2009*. Tübingen : Gunter Narr Verlag, 2009, 53–59
(s. Seite 45)
- [BYRN99] BAEZA-YATES, Ricardo ; RIBEIRO-NETO, Berthier: *Modern Information Retrieval*. 2. Auflage. ACM Press, Addison Wesley, 1999
(s. Seite 95)
- [CAH12] CIGLAN, Marek ; AVERBUCH, Alex ; HLUCHÝ, Ladislav: Benchmarking traversal operations over graph databases. In: *Workshop Proceedings of the 28th International Conference on Data Engineering (ICDE): Third International Workshop on Graph Data Management: Techniques and Applications (GDM)*, 2012, S. 186–189
(s. Seite 187)
- [Cav12] CAVANAGH, Sheila: Living in a Digital World: Rethinking Peer Review, Collaboration, and Open Access. In: *Journal of Digital Humanities* 1 (2012), Nr. 4, S. –
(s. Seite 4)
- [CCM16] CELANO, Giuseppe G. A. ; CRANE, Gregory ; MAJIDI, Saeed: Part of Speech Tagging for Ancient Greek. In: *Open Linguistics* 2 (2016), Nr. 1, S. 393–399
(s. Seite 128)
- [Che76] CHEN, Peter Pin-Shan: The Entity-Relationship Model – Toward a Unified View of Data. In: *ACM Transactions on Database Systems* 1 (1976), Mar, Nr. 1, S. 9–36
(s. Seiten 47 und 231)

- [CJBS16] CHEEMA, Muhammad F. ; JÄNICKE, Stefan ; BLUMENSTEIN, Judith ; SCHEUERMANN, Gerik: A Directed Concept Search Environment to Visually Explore Texts Related to User-defined Concept Models. In: *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) – Volume 2: Information Visualization Theory and Applications (IVAPP)*, 2016, S. 74–85 (s. Seite 188)
- [Cle15] CLEMINSON, Ralph M.: Encoding Text and Encoding Texts: Some Reflections on Theory and Practice. In: *Comparative Oriental Manuscript Studies (COMS) Bulletin* 1 (2015), Nr. 2, S. 77–84 (s. Seite 27)
- [CMBT02] CUNNINGHAM, Hamish ; MAYNARD, Diana ; BONTCHEVA, Kalina ; TABLAN, Valentin: GATE: A framework and graphical development environment for robust NLP tools and applications. In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002, S. 168–175 (s. Seite 232)
- [Cod74] CODD, Edgar F.: Recent Investigations into Relational Data Base Systems. In: ROSENFELD, Jack L. (Hrsg.): *Proceedings of the International Federation for Information Processing (IFIP) Congress Bd. 6, 1974 (IFIP Congress Series)*, S. 1017–1021 (s. Seite 47)
- [CS96] CHRIST, Oliver ; SCHULZE, Bruno M.: Ein flexibles und modulares Anfragesystem für Textcorpora. In: FELDWEIG, Helmut (Hrsg.) ; HINRICHS, Erhard W. (Hrsg.): *Lexikon und Text – Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen*, 1996 (Lexicographica – Supplementbände zum Internationalen Jahrbuch für Lexikographie 73), S. 121–134 (s. Seiten 30 und 230)
- [CSFF16] CIOTTI, Fabio ; SILVIO, Peroni ; FRANCESCA, Tomasi ; FABIO, Vitali: An OWL 2 Formal Ontology for the Text Encoding Initiative. In: *Book of Abstracts of the Digital Humanities Conference*, 2016, S. 151–153 (s. Seite 67)
- [CY15] CRAMER, Catherine ; YAMAMOTO, Eri: *Network Literacy: Essential Concepts and Core Ideas*. Version: Mär 2015. <http://sites.google.com/a/binghamton.edu/netscied/>. Brochure (s. Seite 189)

- [Dam14] DAMEROW, Julia: *A Quadruple-Based Text Analysis System for History and Philosophy of Science*, Arizona State University, Dissertationsschrift (Ph.D.), Aug 2014 (s. Seiten 67 und 193)
- [DCWY16] DAI, Guohao ; CHI, Yuze ; WANG, Yu ; YANG, Huazhong: FPGP: Graph Processing Framework on FPGA – A Case Study of Breadth-First Search. In: *Proceedings of the 24th ACM Special Interest Group on Design Automation (SIGDA) International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2016, S. 105–110 (s. Seite 191)
- [DD94] DEROSE, Steven J. ; DURAND, David G.: *Making Hypermedia Work – A User’s Guide to HyTime*. Kluwer Academic Publisher, 1994 (s. Seite 38)
- [DDL⁺90] DEERWESTER, Scott C. ; DUMAIS, Susan T. ; LANDAUER, Thomas K. ; FURNAS, George W. ; HARSHMAN, Richard A.: Indexing by Latent Semantic Analysis. In: *Journal of the American society for information science (JASIS)* 41 (1990), Nr. 6, S. 391–407 (s. Seite 41)
- [DDMR90] DEROSE, Steven J. ; DURAND, David G. ; MYLONAS, Elli ; RENEAR, Allen H.: What is text, really? In: *J. Computing in Higher Education* 1 (1990), Nr. 2, S. 3–26 (s. Seiten 36 und 234)
- [Dem16] DEMIR, Seniz: Context Tailoring for Text Normalization. In: *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing (TextGraph) at the conference on Human Language Technologies (HLT) of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016, S. 6–14 (s. Seite 95)
- [DFG13] DEUTSCHE FORSCHUNGSGEMEINSCHAFT: *DFG-Praxisregeln „Digitalisierung“ / Deutsche Forschungsgemeinschaft, Förderbereich Literaturversorgungs- und Informationssysteme (LIS)*. 2013 (02/2013) (s. Seiten 5 und 25)
- [DG04] DEAN, Jeffrey ; GHEMAWAT, Sanjay: MapReduce: Simplified Data Processing on Large Clusters. In: *OSDI’04: Sixth Symposium on Operating System Design and Implementation* (2004), 137–150. <http://www.usenix.org/events/osdi04/tech/dean.html> (s. Seite 49)
- [Die16] DIESTEL, Reinhard: *Graduate Texts in Mathematics*. Bd. 173: *Graph Theory*. 5. Edition. Springer-Verlag, Heidelberg, 2016 (s. Seiten 53 und 56)

- [DIPV11] DI IORIO, Angelo ; PERONI, Silvio ; VITALI, Fabio: A Semantic Web approach to everyday overlapping markup. In: *Journal of the Association for Information Science and Technology (AIS&T)* 62 (2011), Nr. 9, S. 1696–1716
(s. Seiten [61](#) und [231](#))
- [DK15] DÖRK, Marian ; KNIGHT, Dawn: WordWanderer: A Navigational Approach to Text Visualisation. In: *Corpora* 10 (2015), Apr, Nr. 1, S. 83–94 (s. Seite [113](#))
- [Doe03] DOERR, Martin: The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata. In: *AI Magazine of the Association for the Advancement of Artificial Intelligence* (2003), S. 75–92
(s. Seite [177](#))
- [DOK⁺16] DROŹDŹ, Stanisław ; OŚWIĘCIMKAA, Paweł ; KULIGA, Andrzej ; KWAPIEŃA, Jarosław ; BAZARNIKB, Katarzyna ; GRABSKA-GRADZIŃSKAC, Iwona ; RYBICKIB, Jan ; STANUSZEKD, Marek: Quantifying origin and character of long-range correlations in narrative texts. In: *Information Sciences* 331 (2016), S. 32–44
(s. Seite [169](#))
- [Dor33] DORNSEIFF, Franz: *Der deutsche Wortschatz nach Sachgruppen*. Walter, 1933
(s. Seite [103](#))
- [Dub04] DUBIN, David: The Most Influential Paper Gerard Salton Never Wrote. In: *Library Trends* 52 (2004), Nr. 4, S. 748–764 (s. Seite [40](#))
- [EAGJ⁺16] EL-ASSADY, Mennatallah ; GOLD, Valentin ; JOHN, Markus ; ERTL, Thomas ; KEIM, Daniel A.: Visual Text Analytics in Context of Digital Humanities. In: *Reports from the 1st IEEE VIS Workshop on Visualization for the Digital Humanities in the Proceedings of the IEEE VIS2016*, 2016, S. im Druck (s. Seite [110](#))
- [EBH15] EFER, Thomas ; BLECHER, Jens ; HEYER, Gerhard: Leipziger Rektoratsreden 1871 - 1933 – Insights into six decades of scientific practice. In: GIPPERT, Jost (Hrsg.) ; GEHRKE, Ralf (Hrsg.): *Historical Corpora. Challenges and Perspectives* Bd. 5, Narr, März 2015 (Korpuslinguistik und interdisziplinäre Perspektiven auf Sprache (CLIP)), S. 229–240 (s. Seite [136](#))

- [Efe15] EFER, Thomas: Text Mining with Graph Databases - Traversal of Persisted Token-Level Representations for Flexible On-Demand Processing. In: UNGER, Herwig (Hrsg.); HALANG, Wolfgang A. (Hrsg.): *Autonomous Systems - Proceedings of the 8th GI Conference*, VDI Verlag, 2015 (Fortschritt-Berichte VDI, Reihe 10 842), S. 157–167 (s. Seite 75)
- [EHJ15] EFER, Thomas ; HEYER, Gerhard ; JOST, Jürgen: Text Mining am Beispiel der Dramen Shakespeares – Welche neuen Erkenntnisse können moderne formale Methoden liefern? In: JANSOHN, Christa (Hrsg.) ; HABICHT, Werner (Hrsg.) ; MEHL, Dieter (Hrsg.) ; REDL, Philipp (Hrsg.) ; Akademie der Wissenschaften und der Literatur, Mainz (Veranst.): *Shakespeare unter den Deutschen* Bd. 2014.3 Akademie der Wissenschaften und der Literatur, Mainz, Franz Steiner Verlag, 2015 (Abhandlungen der Geistes- und Sozialwissenschaftlichen Klasse), S. 217–230 (s. Seiten 161 und 166)
- [EKM15] ECKLE-KOHLER, Judith ; MCCRAE, John P. ; CHIARCOS, Christian: lemonUby – a large, interlinked, syntactically-rich lexical resource for ontologies. In: *Semantic Web* 6 (2015), Nr. 5, S. 371–378 (s. Seite 67)
- [EM16] EGER, Steffen ; MEHLER, Alexander: On the Linearity of Semantic Change: Investigating Meaning Variation via Dynamic Graph Models. In: *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*, 2016, S. 52–58 (s. Seite 66)
- [ESH14] EFER, Thomas ; STEINBACH-HÜTHER, Ninja: Quantitative Analyses in Global and Area Studies using Graph-based Filtering of Heterogeneous Catalogue Data. In: PLÖDEREDER, Erhard (Hrsg.) ; GRUNSKE, Lars (Hrsg.) ; SCHNEIDER, Eric (Hrsg.) ; ULL, Dominik (Hrsg.): *Proceedings of INFORMATIK 2014*. Bonn : Gesellschaft für Informatik, September 2014, S. 1027–1037 (s. Seiten 150 und 159)
- [Fec16] FECHNER, Martin: *Kommunikation von Wissenschaft in der Neuzeit: vom Labor in die Öffentlichkeit*. Max-Planck-Institut für Wissenschaftsgeschichte, 2016 <http://hdl.handle.net/11858/00-001M-0000-002A-C481-2> (s. Seiten 22, 23, 29 und 178)
- [Fel98] FELLBAUM, Christiane (Hrsg.): *WordNet - An Electronic Lexical Database*. 1998 (Language, Speech, and Communication) (s. Seite 67)

- [FFR16] FREIRE, Juliana ; FUHR, Norbert ; RAUBER, Andreas: Reproducibility of Data-Oriented Experiments in e-Science. In: *Dagstuhl Reports* Bd. 6, 2016, S. 108–159 (s. Seite 19)
- [FG06] FUHR, Norbert ; GÖVERT, Norbert: Retrieval Quality vs. Effectiveness of Specificity-Oriented Search in XML Collections. In: *Information Retrieval* 9 (2006), S. 55–70 (s. Seite 108)
- [Fie00] FIELDING, Roy T.: *Architectural Styles and the Design of Network-based Software Architectures*, University of California, Irvine, Diss., 2000. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (s. Seiten 85 und 235)
- [FJ15] FLANDERS, Julia ; JANNIDIS, Fotis: *Knowledge Organization and Data Modeling in the Humanities*. 2015 (s. Seite 23)
- [Fla09] FLANAGAN, David: *MQL Reference Guide*. Metaweb Technologies, 2009 (s. Seite 62)
- [FLM⁺10] FAN, Wenfei ; LI, Jianzhong ; MA, Shuai ; TANG, Nan ; WU, Yinghui ; WU, Yunpeng: Graph Pattern Matching: From Intractable to Polynomial Time. In: *Proceedings of the International Conference on Very Large Data Bases (VLDB)* Bd. 3, 2010 (Proceedings of the VLDB Endowment), S. 264–275 (s. Seite 63)
- [FPT14] FU, Zhisong ; PERSONICK, Michael ; THOMPSON, Bryan: MapGraph: A High Level API for Fast Development of High Performance Graph Analytics on GPUs. In: *Proceedings of the Workshop on GRaph Data management Experiences and Systems (GRADES)*, 2014, S. 6 pp (s. Seite 191)
- [Fri05] FRITZ, Gerd: *Einführung in die historische Semantik*. De Gruyter, 2005 (Germanistische Arbeitshefte 42) (s. Seite 103)
- [Fri06] FRITZ, Gerd: *Historische Semantik*. 2. Auflage. J. B. Metzler, 2006 (Sammlung Metzler) (s. Seite 102)
- [GEKH⁺12] GUREVYCH, Iryna ; ECKLE-KOHLER, Judith ; HARTMANN, Silvana ; MATUSCHEK, Michael ; MEYER, Christian M. ; WIRTH, Christian: UBY – A Large-Scale Unified Lexical-Semantic Resource Based on LMF. In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2012, S. 580–590 (s. Seite 67)

- [GLN⁺94] GIBBONS, Michael ; LIMOGES, Camille ; NOWOTNY, Helga ; SCHWARTZMAN, Simon ; SCOTT, Peter ; TROW, Martin: *The New Production of Knowledge – The Dynamics of Science and Research in Contemporary Societies*. SAGE Publications, 1994 (s. Seite 18)
- [GMPPM13] GLUSHKO, Robert J. ; MAYERNIK, Matthew ; PEPE, Alberto ; MALONEY, Murray: Describing Relationships and Structures. In: GLUSHKO, Robert J. (Hrsg.): *The Discipline of Organizing*. The MIT Press, 2013, S. 189–234 (s. Seite 35)
- [Gol12] GOLD, Matthew K.: The Digital Humanities Moment. In: GOLD, Matthew K. (Hrsg.): *Debates in the Digital Humanities*. University of Minnesota Press, 2012, S. ix ff. (s. Seite 17)
- [Gör11] GÖRZ, Günther: WissKI - Semantische Annotation, Wissensverarbeitung und Wissenschaftskommunikation in einer virtuellen Forschungsumgebung. In: *Kunstgeschichte. Open Peer Reviewed Journal* (2011), S. 1–17 (s. Seite 190)
- [GR10] GALEY, Alan ; RUECKER, Stan: How a prototype argues. In: *Literary and Linguistic Computing (LLC)* 25 (2010), Nr. 4, S. 405–424 (s. Seite 74)
- [Gra73] GRANOVETTER, Mark S.: The Strength of Weak Ties. In: *American Journal of Sociology* 78 (1973), May, Nr. 6, S. 1360–1380 (s. Seite 51)
- [Gra14] GRANDJEAN, Martin: La connaissance est un réseau – Perspective sur l’organisation archivistique et encyclopédique. In: *Les Cahiers du numérique - Les humanités délivrées* 10 (2014), Nr. 3, S. 37–54 (s. Seiten 139 und 140)
- [Gre14] GREEN, Harriett E.: Under the Workbench: An analysis of the use and preservation of MONK text mining research software. In: *Literary and Linguistic Computing (LLC)* 29 (2014), Nr. 1, S. 23–40 (s. Seite 190)
- [Grz11] GRZEGA, Joachim (Hrsg.): *A Recollection of 11 Years of Onomasiology Online (2000–2010): All Articles Re-Collected*. Kompilation. <http://www1.ku-eichstaett.de/SLF/EngluVglSW/OnOn-Total.pdf>. Stand: 2011 (s. Seite 104)
- [GS12] GRIFFIN, Christopher ; SHEKHAR, Suraj: *Penn State Lecture Notes. Bd. Math 485: Graph Theory*. –, 2012 (s. Seite 53)

- [GSB16] GINDA, Michael ; SCHARNHORST, Andrea ; BÖRNER, Katy: Modelling the Structure and Dynamics of Science Using Books. In: SUGIMOTO, Cassidy (Hrsg.): *Theories of Informetrics: A Festschrift in Honor of Blaise Cronin*. De Gruyter Saur, 2016, S. 304–334 (s. Seite 157)
- [GSBH12] GOLLUB, Tim ; STEIN, Benno ; BURROWS, Steven ; HOPPE, Dennis: TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. In: *Proceedings of the 23rd International Workshop on Database and Expert Systems Applications (DEXA)*, 2012, S. 151–155 (s. Seite 121)
- [GTW13] GOODING, Paul ; TERRAS, Melissa ; WARWICK, Claire: The Myth of the New: Mass Digitization, Distant Reading, and the Future of the Book. In: *Literary and Linguistic Computing (LLC)* 28 (2013), Aug, Nr. 4, S. 629–639 (s. Seite 5)
- [Haa01] HAARMANN, Harald: Sprachtypologie und Schriftgeschichte. In: *Sprachtypologie und sprachliche Universalien* Bd. 20. 1. Halbband. De Gruyter Mouton, 2001 Handbücher zur Sprach- und Kommunikationswissenschaft (HSK), S. 163–180 (s. Seite 32)
- [Har75] HARRIS, Bernard: The statistical estimation of entropy in the non-parametric case. In: *Topics in Information Theory* (1975), S. 323–355 (s. Seite 170)
- [Häu09] HÄUSER, Franz (Hrsg.): *Die Leipziger Rektoratsreden 1871-1933: Herausgegeben zum 600-jährigen Gründungsjubiläum der Universität im Jahr 2009*. Walter de Gruyter, 2009 (s. Seiten 136 und 243)
- [HFBPL09] HEBELER, John ; FISHER, Matthew ; BLACE, Ryan ; PEREZ-LOPEZ, Andrew: *Semantic Web Programming*. Wiley Publishing, 2009 (s. Seiten 59 und 60)
- [HSH15] HENRICH, Andreas ; HEYER, Gerhard ; SCHLIEDER, Christoph ; HÄRDER, Theo: Editorial zum Schwerpunktthema *Informationsmanagement für Digital Humanities*. In: *Datenbank-Spektrum* 2015 (2015), Mär, Nr. 1, S. 1–6 (s. Seite 18)
- [HI14] HEYER, Gerhard ; ISEMAN, Daniel: Digital und Computational Humanities. In: *Book of Abstracts zum DHd Workshop „Informatik und die Digital Humanities“*, 2014, S. C.4 (67–68) (s. Seite 19)
- [Hic15] HICKSON, Ian: *Server-Sent Events* / World Wide Web Consortium. Version: Feb 2015. <http://www.w3.org/TR/2015/REC-eventsource-20150203/>. 2015 (REC-eventsource-20150203) (s. Seite 86)

- [Hin13] HINDLEY, Meredith: The Rise of the Machines – NEH and the Digital Humanities: The Early Years. In: *Humanities* 34 (2013), Nr. 4. <http://www.neh.gov/humanities/2013/julyaugust/feature/the-rise-the-machines>
(s. Seite 15)
- [Hir11] HIRSCH, Brett D.: The Kingdom has been Digitized: Electronic Editions of Renaissance Drama and the Long Shadows of Shakespeare and Print. In: *Literature Compass* 8 (2011), Nr. 9, S. 568–591
(s. Seite 164)
- [HjØ15] HJØRLAND, Birger: Classical databases and knowledge organization: A case for boolean retrieval and human decision-making during searches. In: *Journal of the Association for Information Science and Technology (AIS&T)* 66 (2015), Nr. 8, S. 1559–1575
(s. Seite 98)
- [HN13] HAUSENBLAS, Michael ; NADEAU, Jacques: Apache Drill: Interactive Ad-Hoc Analysis at Scale. In: *Big Data*, Mary Ann Liebert Inc. 1 (2013), Jun, Nr. 2
(s. Seite 50)
- [Hob70] HOBERG, Rudolf: *Die Lehre vom sprachlichen Feld*. Schriften des Instituts für Deutsche Sprache, 1970 (Sprache der Gegenwart 11)
(s. Seite 103)
- [Hoc04] HOCKEY, Susan: The History of Humanities Computing. In: SCHREIBMAN, Susan (Hrsg.) ; SIEMENS, Ray (Hrsg.) ; UNSWORTH, John (Hrsg.): *A Companion to Digital Humanities*. Blackwell Publishing, 2004, S. 3–19
(s. Seite 15)
- [Hof99] HOFMANN, Thomas: Probabilistic Latent Semantic Analysis. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999, S. 289–296
(s. Seite 41)
- [HOH06] HILDEBRAND, Michiel ; OSSENBRUGGEN, Jacco van ; HARDMAN, Lynda: /facet: A Browser for Heterogeneous Semantic Web Repositories. In: CRUZ, Isabel (Hrsg.) ; DECKER, Stefan (Hrsg.) ; ALLEMANG, Dean (Hrsg.): *Proceedings 5th International Semantic Web Conference (ISWC)* Bd. 4273, 2006 (Lecture Notes in Computer Science), S. 272–285
(s. Seite 100)
- [HQQW08] HEYER, Gerhard ; QUASTHOFF, Uwe ; WITTIG, Thomas: *Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse*. 1. korrigierter Nachdruck. W3L-Verlag, 2008
(s. Seite 39)

- [HR83] HAERDER, Theo ; REUTER, Andreas: Principles of Transaction-Oriented Database Recovery. In: *ACM Computing Surveys* 15 (1983), Dec, Nr. 4, S. 287–317
(s. Seiten [47](#) und [229](#))
- [HW05] HUMM, Bernhard ; WIETEK, Frank: Architektur von Data Warehouses und Business Intelligence Systemen. In: *Informatik-Spektrum* (2005), Nr. 23, S. 3–14
(s. Seite [47](#))
- [IS07] IDE, Nancy ; SUDERMAN, Keith: GrAF: A Graph-based Format for Linguistic Annotations. In: *Proceedings of the Linguistic Annotation Workshop (LAW)*. Prague, Czech Republic : Association for Computational Linguistics, June 2007, 1–8
(s. Seite [66](#))
- [ISO2709] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information and documentation – Format for information exchange* / ISO. Stage 90.20. 2016 (2016-07-15)
(s. Seite [232](#))
- [ISO9075] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information technology – Database languages – SQL* / ISO. 2008/2015 (Parts: 1,2,3,4,9,10,11,13,14)
(s. Seite [235](#))
- [ISO13250] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information technology – SGML applications – Topic maps* / ISO. Second Edition, Stage 90.92. 2003 (2003-10-23; ISO/IEC 13250)
(s. Seite [62](#))
- [ISO21127] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information and documentation – A reference ontology for the interchange of cultural heritage information* / ISO. Second Edition, Stage 60.60. 2014 (2014-10-15)
(s. Seite [230](#))
- [ISO24622] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Language resource management – Component Metadata Infrastructure (CMDI) – Part 1: The Component Metadata Model* / ISO. Stage 60.60. 2015 (2015-01-20)
(s. Seite [230](#))
- [ISO24624] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Language resource management – Transcription of spoken language* / ISO. Stage 60.60. 2016 (2016-07-22)
(s. Seite [193](#))
- [ISO32000] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Document management – Portable document format – Part 1: PDF 1.7* / ISO. Stage 90.93. 2008 (2008-07-01)
(s. Seite [234](#))

- [Jac06] JACKSON, Matthew O.: The Economics of Social Networks. In: BLUNDELL, Richard (Hrsg.); NEWAY, Whitney (Hrsg.); PERSSON, Torsten (Hrsg.): *Advances in Economics and Econometrics, Theory and Applications: Ninth World Congress of the Econometric Society* Bd. 1. Cambridge University Press, 2006, S. 1–56 (s. Seite 51)
- [Jäg07] JÄGER, Ludwig: *Ferdinand de Saussure – zur Einführung*. Junius, 2007 (s. Seite 40)
- [JB72] JAUCH, Josef-Maria; BARON, J. G.: Entropy, Information and Szilard's Paradox. In: *Helvetica Physica Acta* 45 (1972), Nr. 2, S. 220–232 (s. Seite 166)
- [JBR⁺17] JÄNICKE, Stefan; BLUMENSTEIN, Judith; RÜCKER, Michaela; ZECKZER, Dirk; SCHEUERMANN, Gerik: Visualizing the Results of Search Queries on Ancient Text Corpora with Tag Pies. In: *Pre-Print* (2017), – (in Veröffentlichung). <http://www.informatik.uni-leipzig.de/~stjaenicke/TagPies.pdf> (s. Seite 110)
- [JEBS15] JÄNICKE, Stefan; EFER, Thomas; BÜCHLER, Marco; SCHEUERMANN, Gerik: Designing Close and Distant Reading Visualizations for Text Re-use. In: *Proceedings of the 9th International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) 2014*, 2015, S. 153–171 (s. Seite 165)
- [JGBS14] JÄNICKE, Stefan; GEßNER, Annette; BÜCHLER, Marco; SCHEUERMANN, Gerik: Visualizations for Text Re-use. In: *Proceedings of the 5th International Conference on Information Visualization Theory and Applications, IVAPP*, 2014, S. 59–70 (s. Seite 165)
- [JGW16] JURISH, Bryan; GEYKEN, Alexander; WERNEKE, Thomas: DiaCollo: diachronen Kollokationen auf der Spur. In: *Konferenzabstracts der Digital Humanities im deutschsprachigen Raum (DHd)*, 2016, S. 172–175 (s. Seiten 67 und 114)
- [JM07] JONES, Michael N.; MEWHORT, Douglas J. K.: Representing Word Meaning and Order Information in a Composite Holographic Lexicon. In: *Psychological Review (American Psychological Association)* 114 (2007), Nr. 1, S. 1–37 (s. Seite 41)
- [Joc13] JOCKERS, Matthew L.; SCHREIBMAN, Susan (Hrsg.); SIEMENS, Raymond C. (Hrsg.): *Macroanalysis - Digital Methods and Literary History*. University Of Illinois Press, 2013 (Topics in the Digital Humanities) (s. Seiten 20, 64 und 169)

- [JPT⁺16] JUNGHANN, Martin ; PETERMANN, André ; TEICHMANN, Niklas ; GÓMEZ, Kevin ; RAHM, Erhard: Analyzing Extended Property Graphs with Apache Flink. In: *Proceedings of the 1st ACM Special Interest Group on Management of Data (SIGMOD) Workshop on Network Data Analytics*, 2016, S. #3 (s. Seite 191)
- [JR13] JOUILI, Salim ; REYNAGA, Aldemar: imGraph: A distributed in-memory graph database. In: *IEEE Proceedings of the 2013 International Conference on Social Computing (SocialCom) via the 28th Academy of Science and Engineering (ASE) International Conference on Big Data* (2013) (s. Seite 80)
- [JV13] JOUILI, Salim ; VANSTEENBERGHE, Valentin: An empirical comparison of graph databases. In: *Proceedings of the IEEE International Conference on Security, Risk and Trust, 2010, 2013*, S. 708–715 (s. Seiten 56 und 81)
- [JVHB14] JACOMY, Mathieu ; VENTURINI, Tommaso ; HEYMANN, Sebastien ; BASTIAN, Mathieu: ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. In: *PLoS ONE* 9 (2014), 06, S. 1–12. <http://dx.doi.org/10.1371/journal.pone.0098679>. DOI 10.1371/journal.pone.0098679 (s. Seiten 114, 162 und 181)
- [KA16] KAUFMANN, Sascha ; ANDREWS, Tara L.: Bearbeitung und Annotation historischer Texte mittels Graph-Datenbanken am Beispiel der Chronik des Matthias von Edessa. In: *Konferenzabstracts der Digital Humanities im deutschsprachigen Raum (DHD)*, 2016, S. 176–178 (s. Seiten 68 und 192)
- [KCH⁺90] KANG, Kyo C. ; COHEN, Sholom G. ; HESS, James A. ; NOVAK, William E. ; PETERSON, A. S.: *Feature-Oriented Domain Analysis (FODA) – Feasibility Study / Software Engineering Institute; Carnegie Mellon University; Domain Analysis Project*. 1990 (Technical Report CMU/SEI-90-TR-21; ESD-90-TR-222) (s. Seite 69)
- [KEAH14] KURAS, Christoph ; EFER, Thomas ; ADAM, Christian ; HEYER, Gerhard: The GDR Through the Eyes of the Stasi – Data Mining on the Secret Reports of the State Security Service of the former German Democratic Republic. In: FRED, Ana (Hrsg.) ; FILIPE, Joaquim (Hrsg.) ; INSTICC (Veranst.): *Proceedings of the 6th KDIR INSTICC*, SCITEPRESS – Science and Technology Publications, Oktober 2014, S. 360–365 (s. Seite 140)

- [Kir10] KIRSCHENBAUM, Matthew: What Is Digital Humanities and What's It Doing in English Departments? In: *ADE Bulletin* Bd. 1. Association of Departments of English (ADE) of the Modern Language Association of America (MLA), 2010, S. 55–61 (s. Seite 17)
- [KKEM10] KEIM, Daniel (Hrsg.); KOHLHAMMER, Jörn (Hrsg.); ELLIS, Geoffrey (Hrsg.); MANSMANN, Florian (Hrsg.): *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010 <http://www.vismaster.eu/> (s. Seite 110)
- [Köh05] KÖHLER, Reinhard: Korpuslinguistik – zu wissenschaftstheoretischen Grundlagen und methodologischen Perspektiven. In: MEHLER, Alexander (Hrsg.): *LDV Forum – Themenschwerpunkt Korpuslinguistik* Bd. 20. Gesellschaft für Linguistische Datenverarbeitung (GLDV), 2005 Zeitschrift für Computerlinguistik und Sprachtechnologie, S. 1–16 (s. Seite 29)
- [Kön16] KÖNIG, Mareike: *Was sind Digital Humanities? Definitionsfragen und Praxisbeispiele aus der Geschichtswissenschaft*. <http://dhdhi.hypotheses.org/2642>. Stand: Feb 2016. Blogpost auf "Digital Humanities am Deutschen Historischen Institut Paris (DIHP)" (s. Seite 17)
- [Kön36] KÖNIG, Dénes: *Theorie der Endlichen und Unendlichen Graphen: Kombinatorische Topologie der Streckenkomplexe*. Akademische Verlagsgesellschaft M. B. H., 1936 (s. Seite 52)
- [Kor06] KORPELA, Jukka K.: *Unicode Explained*. O'Reilly & Associates, 2006 (s. Seite 33)
- [KSM13] KOLOMIČENKO, Vojtěch; SVOBODA, Martin; MLÝNKOVÁ, Irena H.: Experimental Comparison of Graph Databases. In: *Proceedings of the 15th International Conference on Information Integration and Web Based Applications & Services (iiWAS)*, 2013, S. 115–124 (s. Seite 81)
- [Kuc16] KUCZERA, Andreas: Digital Editions beyond XML – Graph-based Digital Editions. In: DÜRING, Marten (Hrsg.); JATOWT, Adam (Hrsg.); PREISER-KAPPELLER, Johannes (Hrsg.); DEN BOSCH, Antal van (Hrsg.): *Proceedings of the 3rd Historinformatics Workshop at the Digital Humanities conference (DH)*, 2016, S. 37–46 (s. Seiten 68 und 75)

- [Lad13] LADWIG, Günter: *Efficient Optimization and Processing of Queries over Text-rich Graph-structured Data*. KIT Scientific Publishing Straße am Forum 2 D-76131 Karlsruhe, Karlsruher Institut für Technologie, Diss., 2013. <http://dx.doi.org/10.5445/KSP/1000034423>. DOI 10.5445/KSP/1000034423 (s. Seite 61)
- [LC13] LAKE, Peter ; CROWTHER, Paul: *Concise Guide to Databases – A Practical Introduction*. Springer, 2013 (Undergraduate Topics in Computer Science) (s. Seite 46)
- [Leh09] LEHNER, Franz: *Wissensmanagement – Grundlagen, Methoden und technische Unterstützung*. Dritte Auflage. Carl Hanser Verlag, 2009 (s. Seite 102)
- [Lei14] LEINSTER, Tom: Rethinking Set Theory. In: *American Mathematical Monthly* 121 (2014), 2014, Nr. 5, S. 403–415 (s. Seite 53)
- [Lev66] LEVENSHTAIN, Vladimir I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In: *Soviet Physics Doklady* 10 (1966), Feb, Nr. 8, S. 707–710 (s. Seite 173)
- [Lew92] LEWIS, David D.: *Representation and Learning in Information Retrieval*, University of Massachusetts, Dissertationsschrift (Ph.D.), Februar 1992 (s. Seite 96)
- [Ley09] LEY, Michael: DBLP – Some Lessons Learned. In: *Proceedings of the International Conference on Very Large Data Bases (VLDB)* Bd. 2, 2009 (Proceedings of the VLDB Endowment 1), S. 1493–1500 (s. Seite 152)
- [LGK⁺12] LOW, Yucheng ; GONZALEZ, Joseph ; KYROLA, Aapo ; BICKSON, Danny ; GUESTRIN, Carlos ; HELLERSTEIN, Joseph M.: Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. In: *Proceedings of the 38th International Conference on Very Large Data Bases (VLDB)* Bd. 5, 2012 (Proceedings of the VLDB Endowment 8), S. 716–726 (s. Seite 192)
- [Li90] LI, Wentian: Mutual Information Functions versus Correlation Functions. In: *Journal of Statistical Physics* 60 (1990), Nr. 5/6, S. 823–837 (s. Seite 166)
- [LIJ⁺15] LEHMANN, Jens ; ISELE, Robert ; JAKOB, Max ; JENTZSCH, Anja ; KONTOKOSTAS, Dimitris ; MENDES, Pablo N. ; HELLMANN, Sebastian ; MORSEY, Mohamed ; KLEEF, Patrick van ; AUER, Sören ; BIZER, Christian: DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. In: *Semantic Web – Interoperability, Usability, Applicability (IOS Press)* 6 (2015), Nr. 2, S. 167–195 (s. Seite 60)

- [Lin12] LIN, Yu wei: Transdisciplinarity and Digital Humanities: Lessons Learned from Developing Text-Mining Tools for Textual Analysis. In: BERRY, David M. (Hrsg.): *Understanding Digital Humanities*. Palgrave Macmillan, 2012, S. 295–314 (s. Seite 18)
- [LM04] LETHBRIDGE, Stefanie ; MILDORF, Jarmila: *Basics for English / Part 3: Drama*. Universites of Tübingen, Stuttgart and Freiburg, 2004 <http://www2.anglistik.uni-freiburg.de/intranet/englishbasics/PDF/Drama.pdf> (s. Seite 161)
- [LPA⁺09] LAZER, David ; PENTLAND, Alex ; ADAMIC, Lada ; ARAL, Sinan ; BARABÁSI, Albert-László ; BREWER, Devon ; CHRISTAKIS, Nicholas ; CONTRACTOR, Noshir ; FOWLER, James ; GUTMANN, Myron ; JEBARA, Tony ; KING, Gary ; MACY, Michael ; ROY, Deb ; ALSTYNE, Marshall V.: Computational Social Science. In: *Science* 323 (2009), Feb, Nr. 5915, S. 721–723 (s. Seite 51)
- [LS08] LIU, Yin ; SMITH, Jeff: A Relational Database Model for Text Encoding. In: *Computing in the Humanities Working Papers (VHWP)* (2008), Jul, S. A.43. (s. Seiten 67 und 193)
- [Luh57] LUHN, Hans P.: A Statistical Approach to Mechanized Encoding and Searching of Literary Information. In: *IBM Journal of Research and Development* 1 (1957), Nr. 4, S. 309–317 (s. Seite 95)
- [LV09] LI, Ming ; VITÁNYI, Paul: *An Introduction to Kolmogorov Complexity and Its Applications*. 3. Auflage. Springer, 2009 (Texts in Computer Science) (s. Seite 160)
- [Mar98] MARCUS, Solomon: Bridging Linguistics and Computer Science, via Mathematics. In: CALUDE, Cristian S. (Hrsg.): *People and Ideas in Theoretical Computer Science*. Springer, 1998, S. 163–176 (s. Seite 28)
- [Mar06] MARCHIONINI, Gary: Exploratory Search: From Finding to Understanding. In: *Communications of the ACM* 49 (2006), Apr, Nr. 4 (s. Seite 23)
- [MB09] MILES, Alistair ; BECHHOFFER, Sean: *SKOS Simple Knowledge Organization System – Recommendation* / World Wide Web Consortium. Version: Aug 2009. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>. 2009 (REC-skos-reference) (s. Seite 235)

- [MBJ⁺98] MADISON, Olivia ; BYRUM, Jr. John ; JOUGUELET, Suzanne ; MCGARRY, Dorothy ; WILLIAMSON, Nancy ; WITT, Maria: *UBCIM publications – New Series*. Bd. 19: *Functional requirements for bibliographic records : final report*. International Federation of Library Associations and Institutions (IFLA) Section on Cataloguing, Study Group on the Functional Requirements for Bibliographic Records. K.G. Saur, 1998 (s. Seite 231)
- [McC13a] MCCARTY, Willard: The essential contradiction. In: *6th International Conference Innovative Information Technologies for Science, Business and Education (IIT)*, 2013, S. 12p (s. Seite 21)
- [McC13b] MCCARTY, Willard: The Future of Digital Humanities Is a Matter of Words. In: *Companion to New Media Dynamics*. Wiley-Blackwell, Feb 2013, Kapitel 1.2, S. 33–52 (s. Seiten 14 und 194)
- [MD85] MITTAL, Sanjay ; DYM, Clive L.: Knowledge Acquisition from Multiple Experts. In: *The AI Magazine* 6 (1985), Nr. 2, S. 32–36 (s. Seite 178)
- [MDGM10] MEYER, Scott M. ; DEGENER, Jutta ; GIANNANDREA, John ; MICHENER, Barak: Optimizing Schema-Last Tuple-Store Queries in Graphd. In: *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD) International Conference on Management of Data* (2010), S. 1047–1056 (s. Seiten 61 und 62)
- [Meh09] MEHLER, Alexander: Structure Formation in the Web. In: WITT, Andreas (Hrsg.) ; METZING, Dieter (Hrsg.): *Linguistic Modeling of Information and Markup Languages – Contributions to Language Technology* Bd. 40. Springer, 2009 Text, Speech and Language Technology, S. 225–247 (s. Seite 38)
- [MEP⁺14] MCCOLL, Robert ; EDIGER, David ; POOVEY, Jason ; CAMPBELL, Dan ; BADER, David A.: A Performance Evaluation of Open Source Graph Databases. In: *Proceedings of the first workshop on Parallel programming for analytics applications (PPAA)*, ACM, 2014, S. 11–18 (s. Seite 81)
- [MES07] MELLI, Gabor ; ESTER, Martin ; SARKAR, Anoop: Recognition of Multi-sentence n-ary Subcellular Localization Mentions in Biomedical Abstracts. In: *Short Paper Proceedings of the 2nd International Symposium on Languages in Biology and Medicine (LBM)*, 2007, S. – (s. Seite 66)

- [MGWD10] MEHLER, Alexander ; GLEIM, Rüdiger ; WALTINGER, Ulli ; DIEWALD, Nils: Time Series of Linguistic Networks by Example of the Patrologia Latina. In: *Proceedings of INFORMATIK 2010: Service Science*, 2010, S. 609–616 (s. Seite 66)
- [MH04] MCGUINNESS, Deborah L. ; HARMELEN, Frank van: *OWL Web Ontology Language Overview / World Wide Web Consortium*. Version: Feb 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>. 2004 (REC-owl-features-20040210/) (s. Seite 234)
- [MH12] MCENERY, Tony ; HARDIE, Andrew: *Corpus Linguistics: Method, theory and practice*. Cambridge University Press, 2012 (s. Seite 29)
- [Mic16] MICHELSON, David A.: Syriaca.org as a Test Case for Digitally Re-Sorting the Ancient World. In: CLIVAZ, Claire (Hrsg.) ; DILLEY, Paul (Hrsg.) ; HAMIDović, David (Hrsg.): *Ancient Worlds in Digital Culture*. Koninklijke Brill NV, 2016 Digital Biblical Studies. (Digital Biblical Studies), S. 59–85 (s. Seite 5)
- [Mil55] MILLER, George A.: Note on the bias of information estimates. In: *Information Theory in Psychology: Problems and Methods* (1955) (s. Seite 167)
- [Mil67] MILGRAM, Stanley: The small-world problem. In: *Psychology Today* 1 (1967), May, Nr. 1, S. 61–67 (s. Seite 50)
- [MMG99] MIKHEEV, Andrei ; MOENS, Marc ; GROVER, Claire: Named Entity recognition without gazetteers. In: *Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics*. Morristown, NJ, USA : Association for Computational Linguistics, 1999, S. 1–8 (s. Seite 137)
- [Moo51] MOOERS, Calvin N.: *Making Information Retrieval Pay / Zator Company*. 1951 (55) (s. Seite 95)
- [Mor05] MORETTI, Franco: *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso, 2005 (s. Seite 20)
- [Mor11] MORETTI, Franco: Network Theory, Plot Analysis. In: *New Left Review* (2011), S. 80–102 (s. Seite 161)
- [Mor13] MORETTI, Franco: *Distant Reading*. Verso, 2013 (s. Seite 20)
- [MRS08] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *Introduction to Information Retrieval*. Cambridge University Press, 2008 (s. Seite 95)

- [MS99] MANNING, Christopher D. ; SCHÜTZE, Hinrich: *Foundations of Statistical Natural Language Processing*. 1. The MIT Press, 1999 <http://www.worldcat.org/isbn/0262133601>. ISBN 0262133601 (s. Seiten 24 und 97)
- [MS04] MIHOV, Stoyan ; SCHULZ, Klaus U.: Fast Approximate Search in Large Dictionaries. In: *Computational Linguistics* 30 (2004), Nr. 4, S. 451–477 (s. Seite 177)
- [MT04] MIHALCEA, Rada ; TARAU, Paul: TextRank: Bringing Order into Texts. In: LIN, Dekang (Hrsg.) ; WU, Dekai (Hrsg.): *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Jul 2004, S. 404–411 (s. Seiten 65 und 189)
- [Nel93] NELSON, Theodor H.: *Literary Machines : The report on, and of, Project Xanadu concerning word processing, electronic publishing, hypertext, thinkertoys, tomorrow's intellectual revolution, and certain other topics including knowledge, education and freedom*. Edition 93.1. Mindful Press, 1993 (s. Seite 38)
- [Neu15] NEUMANN, Arne: discoursegraphs: A graph-based merging tool and converter for multilayer annotated corpora. In: MEGYESI, Beáta (Hrsg.): *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA)*, Linköping University Electronic Press, 2015, S. 309–312 (s. Seite 66)
- [Nie11] NIELSEN, Michael: *Reinventing Discovery – The New Era of Networked Science*. Princeton University Press, 2011 (s. Seite 3)
- [NRR⁺12] NELSON, Brent ; RUECKER, Stan ; RADZIKOWSKA, Milena ; SINCLAIR, Stéfan ; BROWN, Susan ; BIEBER, Mark: A Short History and Demonstration of the Dynamic Table of Contexts. In: *Scholarly and Research Communication* 3 (2012), Nr. 4, S. 14 pp (s. Seite 78)
- [Nun09] NUNBERG, Geoffrey: Google's Book Search: A Disaster for Scholars. In: *The Chronicle of Higher Education* (2009), Aug. <http://www.chronicle.com/article/Googles-Book-Search-A/48245/> (s. Seite 44)
- [NYSC00] NARAIN, Rakesh ; YADAV, R.C. ; SARKIS, Joseph ; CORDEIRO, James J.: The strategic implications of flexibility in manufacturing systems. In: *International Journal of Agile Management Systems*, 2 (2000), Nr. 3, S. 202–213 (s. Seite 69)

- [OWL09a] OWL WORKING GROUP: *OWL 2 Web Ontology Language: Document Overview* / World Wide Web Consortium. Version: Oktober 2009. <http://www.w3.org/TR/owl2-overview/>. W3C Recommendation, Oktober 2009 (REC-owl2-overview-20121211) (s. Seite 234)
- [OWL09b] OWL WORKING GROUP ; HITZLER, Pascal (Hrsg.) ; KRÖTZSCH, Markus (Hrsg.) ; PARSIA, Bijan (Hrsg.) ; PATEL-SCHNEIDER, Peter F. (Hrsg.) ; RUDOLPH, Sebastian (Hrsg.): *OWL 2 Web Ontology Language: Primer* / World Wide Web Consortium. Version: Oktober 2009. <http://www.w3.org/TR/owl2-primer/>. W3C Recommendation, Oktober 2009 (REC-owl2-primer-20121211) (s. Seite 234)
- [PB86] PAHL, Gerhard ; BEITZ, Wolfgang: *Konstruktionslehre*. Zweite Auflage. Springer-Verlag, 1986 (s. Seite 131)
- [PBMW99] PAGE, Lawrence ; BRIN, Sergey ; MOTWANI, Rajeev ; WINOGRAD, Terry: *The PageRank Citation Ranking: Bringing Order to the Web* / Stanford InfoLab. 1999 (1999-66) (s. Seite 65)
- [Pio12] PIOTROWSKI, Michael ; HIRST, Graeme (Hrsg.): *Synthesis Lectures on Human Language Technology*. Bd. Lecture #17: *Natural Language Processing for Historical Texts*. Morgan & Claypool Publishers, 2012 (s. Seite 102)
- [Pit15] PITAS, Ioannis: *Graph-Based Social Media Analysis*. Chapman and Hall / CRC Press Taylor & Francis, 2015 (Data Mining and Knowledge Discovery) (s. Seite 192)
- [Pri08] PRITCHETT, Dan: In partitioned databases, trading some consistency for availability can lead to dramatic improvement in scalability. In: *ACM Queue* 6 (2008), Jul, Nr. 3, S. 48–55 (s. Seite 49)
- [Pur02] PURCHASE, Helen C.: Metrics for Graph Drawing Aesthetics. In: *Journal of Visual Languages & Computing* 13 (2002), Nr. 5, 501–516. <http://dx.doi.org/10.1006/jvlc.2002.0232>. DOI 10.1006/jvlc.2002.0232. ISSN 1045–926X (s. Seite 114)
- [QFH12] QUASTHOFF, Uwe (Hrsg.) ; FIEDLER, Sabine (Hrsg.) ; HALLSTEINSDÓTTIR, Erla (Hrsg.): *Frequency Dictionaries*. Bd. 1: *Frequency Dictionary German – Häufigkeitswörterbuch Deutsch*. Leipziger Universitätsverlag, 2012 (s. Seite 174)

- [Ram11] RAMSAY, Stephen ; SCHREIBMAN, Susan (Hrsg.) ; SIEMENS, Raymond C. (Hrsg.): *Reading Machines – Towards an Algorithmic Criticism*. University Of Illinois Press, 2011 (Topics in the Digital Humanities) (s. Seiten 20 und 169)
- [RC94] RAO, Ramana ; CARD, Stuart K.: The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 1994, S. 318–322 (s. Seite 153)
- [Ren01] RENEAR, Allen: The descriptive/procedural distinction is flawed. In: *Markup Languages: Theory & Practice 2* (2001), Sep, Nr. 4, S. 411–420 (s. Seite 45)
- [RFC791] POSTEL, Jon: *Internet Protocol – Darpa Internet Program Protocol Specification / Information Sciences Institute, University of Southern California*. Version: Sep 1981. <http://www.rfc-editor.org/rfc/rfc791.txt>. 1981 (791) (s. Seite 232)
- [RFC1630] BERNERS-LEE, Tim: *Universal Resource Identifiers in WWW / Conseil Européen pour la Recherche Nucléaire (CERN)*. Version: Jun 1994. <http://www.rfc-editor.org/rfc/rfc1630.txt>. 1994 (1630) (s. Seite 236)
- [RFC2616] FIELDING, Roy T. ; GETTYS, Jim ; MOGUL, Jeffrey ; NIELSEN, Henrik F. ; MASINTER, Larry ; LEACH, Paul ; BERNERS-LEE, Tim: *Hypertext Transfer Protocol – HTTP/1.1*. Version: June 1999. <http://www.rfc-editor.org/rfc/rfc2616.txt>. 1999 (2616) (s. Seite 232)
- [RGP⁺12] RIEHMANN, Patrick ; GRUENDL, Henning ; POTTHAST, Martin ; TRENMANN, Martin ; STEIN, Benno ; FROELICH, Bernd: WordGraph: Keyword-in-Context Visualization for Netspeak’s Wildcard Search. In: *IEEE Transactions on Visualization and Computer Graphics* (2012), Sep, S. 1411–1423 (s. Seite 113)
- [Rij79] RIJSBERGEN, Cornelis J.: *Information Retrieval*. Butterworth, 1979 (s. Seite 95)
- [Ris14] RISAM, Roopika: Rethinking Peer Review in the Age of Digital Humanities. In: *Ada: A Journal of Gender, New Media, and Technology* (2014), Apr, Nr. 4, S. #6 (s. Seite 4)
- [RLJ99] RAGGETT, Dave ; LE HORS, Arnaud ; JACOBS, Ian: *HTML 4.01 Specification / W3C*. Version: Dez 1999. <http://www.w3.org/TR/html4>. 1999 (401-19991224) (s. Seite 232)

- [RMD96] RENEAR, Allen ; MYLONAS, Elli ; DURAND, David: Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies. In: HOCKEY, Susan (Hrsg.) ; IDE, Nancy (Hrsg.): *Selected Papers from the 1992 Conference of the Association for Computers and the Humanities (ALLC) and the Association for Literary and Linguistic Computing (ACH)*, Clarendon Press, Sep 1996 (Research in Humanities Computing 4), S. 263–280 (s. Seiten 35 und 36)
- [RN11] RODRIGUEZ, Marko A. ; NEUBAUER, Peter: The Graph Traversal Pattern. In: SAKR, Sherif (Hrsg.) ; PARDEDE, Eric (Hrsg.): *Graph Data Management: Techniques and Applications*. Information Science Reference, 2011, Kapitel 2, S. 29–46 (s. Seiten 52, 55, 57, 75 und 174)
- [Rod15] RODRIGUEZ, Marko A.: The Gremlin Graph Traversal Machine and Language. In: *Proceedings of the 15th Symposium on Database Programming Languages*, 2015, S. 1–10 (s. Seiten 63 und 179)
- [Ros03] ROSENZWEIG, Roy: Scarcity or Abundance? Preserving the Past in a Digital Era. In: *American Historical Review* 108 (2003), Jun, Nr. 3, S. 735–762 (s. Seite 6)
- [RRS11] RUECKER, Stan (Hrsg.) ; RADZIKOWSKA, Milena (Hrsg.) ; SINCLAIR, Stéfán (Hrsg.): *Visual Interface Design for Digital Cultural Heritage – A Guide to Rich-Prospect Browsing*. Ashgate/Routledge, 2011 (s. Seite 179)
- [RS16] REPKO, Allen F. ; SZOSTAK, Richard: *Interdisciplinary Research: Process and Theory*. 3. Auflage. SAGE Publications, 2016 (s. Seite 186)
- [Rue15] RUECKER, Stan: A Brief Taxonomy of Prototypes for the Digital Humanities. In: *Scholarly and Research Communication* 6 (2015), Oct, Nr. 2, 1–11. <http://src-online.ca/index.php/src/article/view/222/415> (s. Seite 74)
- [RV13] ROUSSEAU, François ; VAZIRGIANNIS, Michalis: Graph-of-word and TW-IDF - New Approach to Ad Hoc IR. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management (CIKM)*, 2013, S. 59–68 (s. Seiten 66 und 108)
- [RWE13] ROBINSON, Ian ; WEBBER, Jim ; EIFREM, Emil: *Graph Databases*. O'Reilly Media, Inc., 2013. ISBN 1449356265, 9781449356262 (s. Seite 50)

- [RWJ⁺94] ROBERTSON, Stephen E.; WALKER, S.; JONES, S.; HANCOCK-BEAULIEU, M. M.; GATFORD, M.: Okapi at TREC-3. In: *Proceedings of the Third Text REtrieval Conference*, 1994, S. 109–126 (s. Seite 97)
- [SA82] SPENCE, Robert; APPERLEY, Mark: Data base navigation: an office environment for the professional. In: *Behaviour and Information Technology* 1 (1982), Nr. 1, S. 43–54 (s. Seite 111)
- [Sac09] SACCO, Giovanni M.: The Model. In: SACCO, Giovanni M. (Hrsg.); TZITZIKAS, Yannis (Hrsg.): *Dynamic Taxonomies and Faceted Search – Theory, Practice, and Experience*. Springer, 2009, S. 1–17 (s. Seite 99)
- [Sah13] SAHLE, Patrick: *Schriften des Instituts für Dokumentologie und Editorik*. Bd. 9: *Digitale Editionsformen, Zum Umgang mit der Überlieferung unter den Bedingungen des Medienwandels (3): Textbegriffe und Recodierung*. Books on Demand, Norderstedt, 2013 (s. Seite 64)
- [Sah15] SAHLGREN, Magnus: A brief history of word embeddings (and some clarifications). Blogeintrag (2015-09-30). <http://www.gavagai.se/blog/2015/09/30/a-brief-history-of-word-embeddings/>. Stand: 2015 (s. Seite 40)
- [SBdI14] SIMON, Rainer; BARKER, Elton; DE SOTO, Pau; ISAKSEN, Leif: Pelagios. In: ELLIOTT, Thomas (Hrsg.); HEATH, Sebastian (Hrsg.); MUCCIGROSSO, John (Hrsg.): *Current Practice in Linked Open Data for the Ancient World* Bd. 7. Ancient World Digital Library (AWDL), 2014 Institute for the Study of the Ancient World (ISAW) Papers, S. #27 (s. Seite 141)
- [Sch99] SCHWEGMANN, Ansgar; BECKER, Jörg (Hrsg.); GROB, Heinz L. (Hrsg.); KLEIN, Stefan (Hrsg.): *Objektorientierte Referenzmodellierung – Theoretische Grundlagen und praktische Anwendung*. Deutscher Universitätsverlag; Springer, 1999 (Gabler Edition Wissenschaft - Informationsmanagement und Controlling) (s. Seite 178)
- [Sch06] SCHMIDT, Desmond: Graphical Editor for Manuscripts. In: *Literary and Linguistic Computing (LLC)* 21 (2006), Nr. 3, S. 341–351 (s. Seite 67)
- [Sch10] SCHMIDT, Desmond: The inadequacy of embedded markup for cultural heritage texts. In: *Literary and Linguistic Computing (LLC)* 25 (2010), Apr, Nr. 3, S. 337–356 (s. Seiten 37 und 67)

- [Sch14] SCHMIDT, Desmond: Towards an Interoperable Digital Scholarly Edition. In: *Journal of the Text Encoding Initiative* (2014), Nov, Nr. 7 (s. Seite 45)
- [SEB13] SCHARLOTH, Joachim ; EUGSTER, David ; BUBENHOFER, Noah: Das Wuchern der Rhizome. Linguistische Diskursanalyse und Data-driven Turn. In: BUSSE, Dietrich (Hrsg.) ; TEUBERT, Wolfgang (Hrsg.): *Linguistische Diskursanalyse: neue Perspektiven*. Springer VS Verlag, 2013 Interdisziplinäre Diskursforschung. (Interdisziplinäre Diskursforschung), S. 345–380 (s. Seite 16)
- [SET09] SEGARAN, Roby ; EVANS, Colin ; TAYLOR, Jamie: *Programming the Semantic Web – Build Flexible Applications with Graph Data*. O'Reilly Media, Inc., 2009 (s. Seite 61)
- [SFSK15] SUN, Wen ; FOKOUE, Achille ; SRINIVAS, Kavitha ; KEMENTSIETSIDIS, Anastasios: SQLGraph: An Efficient Relational-Based Property Graph Store. In: *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD) International Conference on Management of Data*, 2015, S. 1887–1901 (s. Seite 55)
- [Sha48] SHANNON, Claude E.: A Mathematical Theory of Communication. In: *The Bell System Technical Journal* 27 (1948), Juli, Oktober, S. 379–423, 623–656 (s. Seite 166)
- [Sha08] SHARP, Robin: *Principles of Protocol Design*. Springer-Verlag, Berlin, Heidelberg, 2008 (s. Seite 85)
- [Sin03] SINCLAIR, Stéfan: Computer-Assisted Reading - Reconceiving Text Analysis. In: *Literary and Linguistic Computing (LLC)* 18 (2003), Nr. 2, S. 175–184 (s. Seite 65)
- [SJBL16] SHARMA, Aneesh ; JIANG, Jerry ; BOMMANAVAR, Praveen ; LIN, Jimmy: Graph-Jet: Real-Time Content Recommendations at Twitter. In: *Proceedings of the International Conference on Very Large Data Bases (VLDB) Bd. 9*, 2016 (Proceedings of the VLDB Endowment 13), S. 1281–1292 (s. Seite 191)
- [SK12] SPIVAK, David I. ; KENT, Robert E.: Ologs: A Categorical Framework for Knowledge Representation. In: *PLoS ONE* (2012) (s. Seite 54)

- [SL16] STULPE, Alexander ; LEMKE, Matthias: Blended Reading – Theoretische und praktische Dimensionen der Analyse von Text und sozialer Wirklichkeit im Zeitalter der Digitalisierung. In: LEMKE, Matthias (Hrsg.) ; WIEDEMANN, Gregor (Hrsg.): *Text Mining in den Sozialwissenschaften – Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*. Springer VS Verlag, 2016, S. 17–61 (s. Seite 20)
- [SLK⁺14] SPORNY, Manu ; LONGLEY, Dave ; KELLOGG, Gregg ; LANTHALER, Markus ; LINDSTRÖM, Niklas: *JSON-LD 1.0 – A JSON-based Serialization for Linked Data / W3C*. Version: Jan 2014. <http://www.w3.org/TR/json-ld/>. 2014 (401-19991224) (s. Seiten 61 und 190)
- [SM99] SHIPMAN, Frank M. III ; MARSHALL, Catherine C.: Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. In: *Computer-Supported Cooperative Work* 8 (1999), Nr. 4, S. 333–352 (s. Seiten 133 und 178)
- [SMHR00] SPERBERG-MCQUEEN, C. M. ; HUITFELDT, Claus ; RENEAR, Allen: Meaning and interpretation of markup. In: *Markup Languages: Theory & Practice* 2 (2000), Nr. 3 (s. Seite 60)
- [Spi03] SPINRAD, Jeremy P.: *Fields Institute monographs*. Bd. 19: *Efficient Graph Representations*. American Mathematical Society, 2003 (s. Seite 54)
- [Sri11] SRINIVASA, Srinath: Data, Storage and Index Models for Graph Databases. In: SAKR, Sherif (Hrsg.) ; PARDEDE, Eric (Hrsg.): *Graph Data Management: Techniques and Applications*. Information Science Reference, 2011, Kapitel 3, S. 47–70 (s. Seite 80)
- [SS09] STEDE, Manfred ; SURIYAWONGKUL, Arthit: Identifying Logical Structure and Content Structure in Loosely-Structured Documents. In: WITT, Andreas (Hrsg.) ; METZING, Dieter (Hrsg.): *Linguistic Modeling of Information and Markup Languages – Contributions to Language Technology* Bd. 40. Springer, 2009 Text, Speech and Language Technology, S. 81–96 (s. Seite 35)

- [SS13] SCHULZE, Matthias ; STOCKMANN, Ralf: Open Science und Networked Science – Offenheit und Vernetzung als Leitmotive und Visionen einer digitalen Wissenschaft im 21. Jahrhundert. In: NEUROTH, Heike (Hrsg.) ; LOSSAU, Norbert (Hrsg.) ; RAPP, Andrea (Hrsg.): *Evolution der Informationsinfrastruktur – Kooperation zwischen Bibliothek und Wissenschaft*. Verlag Werner Hülsbusch, 2013, S. 31–38 (s. Seite 3)
- [SSB16] STUTZ, Philip ; STREBEL, Daniel ; BERNSTEIN, Abraham: Signal/collect12: processing large graphs in seconds. In: *Semantic Web – Interoperability, Usability, Applicability (IOS Press)* 7 (2016), Nr. 2, S. 139–166 (s. Seite 192)
- [SSU04] SCHREIBMAN, Susan (Hrsg.) ; SIEMENS, Ray (Hrsg.) ; UNSWORTH, John (Hrsg.): *A Companion to Digital Humanities*. Oxford : Blackwell Publishing, 2004 <http://www.digitalhumanities.org/companion/> (s. Seite 17)
- [Sve12] SVENSSON, Patrik: Envisioning the Digital Humanities. In: *Digital Humanities Quarterly (DHQ)* 6 (2012), Nr. 1, S. #3 (s. Seite 17)
- [SW09] SMITH, D. N. ; WEAVER, Gabriel A.: *Applying Domain Knowledge from Structured Citation Formats to Text and Data Mining: Examples Using the CITE Architecture / Department of Computer Science - Dartmouth College*. 2009 (TR2009-649) (s. Seite 28)
- [SWMM13] SIMON, Agnès ; WENZ, Romain ; MICHEL, Vincent ; MASCIO, Adrien D.: Publishing Bibliographic Records on the Web of Data: Opportunities for the BnF (French National Library). In: *Proceedings of the 10th European Semantic Web Conference (ESWC) – The Semantic Web: Semantics and Big Data* Bd. 7882, 2013 (Lecture Notes in Computer Science), S. 563–577 (s. Seite 159)
- [SWX12] SHAO, Bin ; WANG, Haixun ; XIAO, Yanghua: Managing and Mining Large Graphs: Systems and Implementations. In: *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD) International Conference on Management of Data*, 2012, S. 589–592 (s. Seite 192)
- [SWY75] SALTON, Gerard ; WONG, A. ; YANG, C. S.: A Vector Space Model for Automatic Indexing. In: *Communications of the ACM* 18 (1975), Nov, Nr. 11, S. 613–620 (s. Seiten 97 und 112)

- [TEIP5] TEI CONSORTIUM (Hrsg.): *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative, 2016 (Version 3.0.0). <http://www.tei-c.org/Guidelines/P5/> (s. Seiten 37 und 235)
- [TKG15] TREERATPITUK, Pucktada ; KHABSA, Madian ; GILES, C. L.: Automatically Generating a Concept Hierarchy with Graphs. In: *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2015, S. 265–266 (s. Seite 177)
- [TMK⁺06] TUMMARELLO, Giovanni ; MORBIDONI, Christian ; KEPLER, Fabio N. ; PIAZZA, Francesco ; PULITI, Paolo: A novel Textual Encoding paradigm based on Semantic Web tools and semantics. In: *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC)*, 2006, S. 247–252 (s. Seite 67)
- [TP10] TURNEY, Peter D. ; PANTEL, Patrick: From Frequency to Meaning: Vector Space Models of Semantics. In: *Journal of Artificial Intelligence Research* (2010), Nr. 37, S. 141–188 (s. Seite 40)
- [Tri73] TRIER, Jost: Sprachliche Felder. In: LEE, Anthony van d. (Hrsg.) ; REICHMANN, Oskar (Hrsg.): *Aufsätze und Vorträge zur Wortfeldtheorie*. Posthume Gesamtausgabe. Mouton, 1973, S. 93–109 (s. Seite 103)
- [Tri06] TRIPPEL, Thorsten: *The Lexicon Graph Model: A generic model for multimodal lexicon development*, Universität Bielefeld, Dissertationsschrift, 2006 (s. Seite 67)
- [TWD⁺13] THIYAGALINGAM, Jeyarajan ; WALTON, Simon ; DUFFY, Brian ; TREFETHEN, Anne ; CHEN, Min: Complexity Plots. In: PREIM, Bernhard (Hrsg.) ; RHEINGANS, Penny (Hrsg.) ; THEISEL, Holger (Hrsg.): *Proceedings of the Eurographics Conference on Visualization (EuroVis) 2013* Bd. 32, 2013 (Eurographics Conference on Visualization 3), S. 111–120 (s. Seite 118)
- [Uns00] UNSWORTH, John: Scholarly Primitives: what methods do humanities researchers have in common, and how might our tools reflect this? In: *Symposium on "Humanities Computing: formal methods, experimental practice"*, King's College, London (2000). <http://jefferson.village.virginia.edu/~jmu2m/Kings.5-00/primitives.html> (s. Seite 22)

- [VGB61] VERHOEFF, Jacobus ; GOFFMAN, William ; BELZER, Jack: Inefficiency of the Use of Boolean Functions for Information Retrieval Systems. In: *Communications of the ACM* 4 (1961), Nr. 12, 557–558. <http://dx.doi.org/10.1145/366853.366861>. DOI 10.1145/366853.366861 (s. Seite 98)
- [VMZ⁺10] VICKNAIR, Chad ; MACIAS, Michael ; ZHAO, Zhendong ; NAN, Xiaofei ; CHEN, Yixin ; WILKINS, Dawn: A Comparison of a Graph Database and a Relational Database – A Data Provenance Perspective. In: *Proceedings of the 48th Annual Southeast Regional Conference (SE) of the Association for Computing Machinery (ACM)* (2010), S. #42 (s. Seite 193)
- [WAB⁺09] WITTENBURG, Peter ; AGATONOVICH, Milan ; BEL, Nuria ; BEL, Santi ; BÜCHLER, Marco ; CRISTEA, Dan ; FRITZINGER, Fabienne ; HINRICHS, Erhard ; HINRICHS, Marie ; KEMPS-SNIJDERS, Radu Ionand M. ; RODRIGUEZ, Yana Panchenkoand V. ; SCHMID, Helmut ; QUASTHOFF, Uwe ; VILLEGAS, Martha ; ZASTROW, Thomas: *Requirements Specification Web Services and Workflow Systems* / CLARIN. Version: Jun 2009. <http://www.clarin.eu/files/wg2-6-requirements-doc-v2.pdf>. 2009 (CLARIN-2009-1) (s. Seite 26)
- [Wit04] WITT, Andreas: Linguistische Informationsmodellierung mit XML. In: MEHLER, Alexander (Hrsg.) ; LOBIN, Henning (Hrsg.): *Automatische Textanalyse: Systeme und Methoden zur Annotation und Analyse Natürlichsprachiger Texte*. VS Verlag für Sozialwissenschaften, 2004, S. 39–54 (s. Seite 36)
- [Woo12] WOOD, Peter T.: Query Languages for Graph Databases. In: *ACM Special Interest Group on Management of Data (SIGMOD) Record* 41 (2012), Nr. 1, S. 50–60 (s. Seite 62)
- [Yau13] YAU, Nathan ; LOWE, Jen (Hrsg.): *Data Points – Visualization that Means Something*. Ex. 01. John Wiley & Sons, 2013 (s. Seite 104)
- [ZAB⁺12] ZUNDERT, Joris van ; ANTONIJEVIC, Smiljana ; BEAULIEU, Anne ; DALEN-OSKAM, Karina van ; ZELDENRUST, Douwe ; ANDREWS, Tara L.: Cultures of Formalisation: Towards an Encounter between Humanities and Computing. In: BERRY, David M. (Hrsg.): *Understanding Digital Humanities*. Palgrave Macmillan, 2012, S. 279–294 (s. Seite 188)

- [Zac77] ZACHARY, Wayne W.: An Information Flow Model for Conflict and Fission in Small Groups. In: *Journal of Anthropological Research* 33 (1977), Nr. 4, S. 452–473
(s. Seiten 51, 130 und 180)
- [ZCYL08] ZOU, Lei ; CHEN, Lei ; YU, Jeffrey X. ; LU, Yansheng: A Novel Spectral Coding in a Large Graph Database. In: *Proceedings of the 11th International Conference on Extending Database Technology (EDBT) Advances in Database Technology* (2008), S. 181–192
(s. Seite 54)
- [Zie05] ZIEM, Alexander: *Frame-Semantik und Diskursanalyse – Zur Verwandtschaft zweier Wissensanalysen*. http://www.phil-fak.uni-duesseldorf.de/fileadmin/Redaktion/Institute/Germanistik/Konstruktionsgrammatik/ZiemFrames_Diskurs.pdf. Stand: Jun 2005. Einreichung für die Konferenz Diskursanalyse in Deutschland und Frankreich. Aktuelle Tendenzen in den Sozial- und Sprachwissenschaften
(s. Seite 103)
- [Zip32] ZIPF, George K.: *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press, 1932
(s. Seite 116)
- [ZR10] ZIPSER, Florian ; ROMARY, Laurent: A model oriented approach to the mapping of annotation formats using standards. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC) Workshop on Language Resource and Language Technology Standards (LRLTS)*, 2010, S. 7–18
(s. Seite 66)
- [Zwe16] ZWEIG, Katharina A.: Are Word-Adjacency Networks Networks? In: MEHLER, Alexander (Hrsg.) ; LÜCKING, Andy (Hrsg.) ; BANISCH, Sven (Hrsg.) ; BLANCHARD, Philippe (Hrsg.) ; JOB, Barbara (Hrsg.): *Towards a Theoretical Framework for Analyzing Complex Linguistic Networks*. Springer, 2016 *Understanding Complex Systems*. (Understanding Complex Systems), S. 153–163
(s. Seite 189)

Abkürzungsverzeichnis

ACID	Atomicity, Consistency, Isolation, Durability, <i>Wesentliche Grundanforderungen an relationale Datenbanksysteme, eingeführt in [HR83].</i>
ADHO	Alliance of Digital Humanities Organizations, <i>Internationaler Dachverband von Regionalen Vereinigungen für die Digitalen Geisteswissenschaften. – Webseite: http://adho.org/.</i>
AJAX	Asynchronous JavaScript and XML , <i>Eine weit verbreitete Technik, um über eine Serie von Requests eine asynchrone Kommunikation zwischen Webbrowser und Webserver zu erreichen, während eine Webseite betrachtet wird.</i>
ANNIE	A Nearly-New Information Extraction System, <i>Eine NER-Komponente des GATE-Frameworks.</i>
API	Application Programming Interface, <i>Die Gesamtheit aller Funktionen, die ein Softwareprodukt anderen Programmen zur Verfügung stellt, sowie die Bezeichnung der dafür geschaffenen Schnittstelle.</i>
BASE	Basically Available, Soft state, Eventually consistent, <i>Praxisrelevante Eigenschaften eines verteilten datenhaltenden Systems, welche einen Kompromiss zwischen den Extremen des CAP-Theorems darstellen.</i>
BnF	Bibliothèque nationale de France, <i>Die Französische Nationalbibliothek, mit einem Sammelauftrag, u. a. für alle französischsprachigen sowie alle in Frankreich verlegten Publikationen. – Webseite: http://www.bnf.fr/.</i>
BPEL	(Web Services) Business Process Execution Language, <i>Eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen auf der Grundlage von Webservices. – Webseite: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.</i>

CAP	Consistency, Availability, Partition tolerance, <i>Die einander beeinflussenden Aspekte der Konsistenz, Verfügbarkeit und Partitionstoleranz von verteilten (Datenbank-)Systemen.</i>
CIDOC	International Committee for Documentation of the International Council of Museums, <i>Ein internationaler Verbund zur Weiterentwicklung von Dokumentationsmöglichkeiten für Museumssammlungen. – Webseite: http://cidoc.icom.museum/.</i>
CITE	Collections, Indices, Texts, and Extensions, <i>Eine vom "Homer Multitext Project" konzipierte Sammlung von Protokollen und Formaten für das Verwalten, Austauschen und wissenschaftliche Zitieren von (primär textuellen) Quellen. – Webseite: http://www.homermultitext.org/hmt-doc/cite/.</i>
CLARIN	Common Language Resources and Technologies Infrastructure, <i>Ein europäisches Projekt zur Schaffung einer Forschungs-Infrastruktur für geisteswissenschaftliche Quellen, Sprachressourcen und -technologien sowie Werkzeuge und Services. – Webseite: http://www.clarin.eu/.</i>
CMDI	Component Metadata Infrastructure, <i>Ein von CLARIN initiiertes Standard für die Definition rekombinierbarer wiederverwendbarer Vorlagen für Metadatensätze [ISO24622].</i>
CQL	Corpus Query Language, <i>Eine Abfragesprache für linguistisch annotierte Korpora, beschrieben (noch ohne diesen Namen) z. B. in [CS96].</i>
CREOLE	a Collection of REusable Objects for Language Engineering, <i>Der Verbund von Komponentenbibliothek, Typhierarchie und Plugin-Architektur in GATE.</i>
CRM	Conceptual Reference Model, <i>Eine Referenzontologie der CIDOC für den Austausch im Bereich des Kulturellen Erbes, [ISO21127].</i>
CRUD	Create, Read, Update, Delete, <i>Grundlegende Operationen für die Interaktion mit Datenbanksystemen.</i>
CSS	Cascading Style Sheets, <i>Eine regelbasierte Sprache für die Gestaltung von HTML-Dokumenten über Element-Selektoren und Eigenschaften-Wert-Paare.</i>

CTS	Canonical Text Services, <i>Teil der CITE-Infrastruktur, der sich mit der Definition eines Referenzierungsschemas und verschiedener Zugriffsmethoden für hierarchisch strukturierte Texte beschäftigt.</i>
DARIAH	Digital Research Infrastructure for the Arts and Humanities, <i>Ein europäisches Projekt zur Schaffung einer Forschungs-Infrastruktur für geistes- und kunstwissenschaftliche Forschung. – Webseite: http://www.clarin.eu/.</i>
DFG	Deutsche Forschungsgemeinschaft, <i>Die zentrale Organisation zur Forschungsförderung an Hochschulen und öffentlich finanzierten Forschungsinstituten in Deutschland. – Webseite: http://www.dfg.de/.</i>
DOM	Document Object Model, <i>Ein Datenmodell und eine dazugehörige Programmierschnittstelle zum Auslesen und Bearbeiten von hierarchisch gegliederten Dokumenten.</i>
DSL	Domain Specific Language, <i>Eine für einen bestimmten Einsatzzweck konzipierte Beschreibungs- oder Programmiersprache, die oft Syntax und Ausführungsumgebung einer universellen Sprache mittels Makro- oder Funktionsdefinitionen um spezifische Funktionen erweitert.</i>
EARMARK	Extremely Annotational RDF Markup, <i>Ein System zur Modellierung überlappender Textannotationen mittels Semantic-Web-Technologien [DIPV11].</i>
ERM	Entity-Relationship Model, <i>Ein Werkzeug zur abstrahierten Modellierung von Entitäten sowie ihren gegenseitigen Abhängigkeiten innerhalb einer Wissensdomäne [Che76].</i>
ETL	Extract, Transform, Load, <i>Datenextraktion, -umformung und strukturiertes Einlesen bei der Datenintegration, meist im Umfeld von Data Warehouses und Data-Mining-Anwendungen.</i>
FRBR	Functional Requirements for Bibliographic Records, <i>Ein System von Entitäten unterschiedlicher Abstraktionsgrade, die mit entsprechenden Eigenschaften und Relationen für die Beschreibung bibliographischer Angaben genutzt werden können [MBJ⁺98].</i>

GATE	General Architecture for Text Engineering, <i>eine Programmarchitektur und umfangreiche Werkzeugsammlung für die automatische Sprachverarbeitung [CMBT02].</i>
GNU	GNU 's Not Unix, <i>Ein freies und quelloffenes Betriebssystem inklusive einer großen und auch in anderen Betriebssystemen populären Sammlung von Hilfsprogrammen.</i>
HTML	Hypertext Markup Language, <i>Eine SGML-basierte Hypertextsprache, Standarddateiformat im World Wide Web, [RLJ99].</i>
HTTP	Hypertext Transfer Protocol, <i>Ein Internet-Protokoll der Anwendungsschicht zur Anfrage und Übertragung von Hypertext-Dokumenten und anderen Dateien [RFC2616].</i>
ICU	International Components for Unicode, <i>Eine Sammlung von (mittlerweile durch das Unicode-Konsortium gelenkten) Referenzimplementierungen von Unicode- und Globalisierungs-Funktionen.</i>
IP	Internet Protocol, <i>Ein verbindungsloses paketvermittelndes Protokoll, das die Basis der Internetkommunikation bildet [RFC791].</i>
ISO	International Organization for Standardization, <i>Die internationale Dachorganisation nationaler Standardisierungsinstitute und -gremien. – Webseite: http://www.iso.org/.</i>
JAPE	Java Annotation Patterns Engine, <i>Die Musterbeschreibungs-Komponente von GATE.</i>
JSON	JavaScript Object Notation, <i>Ein populäres und leichtgewichtiges Serialisierungs- und Austauschformat auf der Basis von JavaScript-Syntaxelementen.</i>
LOD	Linked Open Data, <i>Ein verteiltes System frei verfügbarer Datensammlungen, die über die entsprechenden Beschreibungs- und Referenzierungsmechanismen des Semantic Web modelliert und miteinander verbunden sind.</i>
MARC	MAchine-Readable Cataloging, <i>Ein Datenformat für bibliographische Katalogeinträge, codiert nach [ISO2709].</i>

MODS	Metadata Object Description Schema, <i>Ein von der Library of Congress entwickeltes Metadatenformat für den Bibliotheks- bereich, das sich an MARC orientiert und Möglichkeiten zur Abbildung zusätzlicher Eigenschaften bietet.</i>
NER	Named Entity Recognition, <i>Eigennamenerkennung, oft einhergehend mit der Einordnung in Typen von Entitäten.</i>
NFC	Normalization Form Canonical Composition, <i>In Unicode die systematische Wahl von sequenziellen Einzel-Kodierungspunkten als Normalisierungsformen für Zeichen, die aus mehreren Komponenten bestehen.</i>
NFD	Normalization Form Canonical Decomposition, <i>In Unicode die systematische Wahl von Kodierungspunkten, die vorkombinierte Zei- chen beschreiben, als Normalisierungsformen für Zeichen, die aus mehreren Kompo- nenten bestehen.</i>
NLP	Natural Language Processing, <i>Verfahren für die digitale Verarbeitung, Strukturierung und Analyse natürlichspra- chiger Textdokumente; zu deutsch: Automatische Sprachverarbeitung.</i>
NoSQL	Not only SQL , <i>Sammelbegriff für Datenbanksysteme, in denen (verglichen mit herkömmlichen rela- tionalen Systemen) alternative Ansätze im Hinblick auf Modellierung, Speicherung und Abfrage der Einträge verfolgt werden.</i>
OAI	Open Archives Initiative, <i>Eine Initiative zur Förderung von standardisierten Austauschmöglichkeiten für digi- tale Inhalte. – Webseite: http://www.openarchives.org/.</i>
OAI-PMH	OAI Protocol for Metadata Harvesting, <i>Ein von der OAI entwickeltes Protokoll inklusive einer Schnittstellenbeschreibung für die automatische Übernahme von Metadaten aus verteilten Quellen. – Webseite: http://www.openarchives.org/pmh/.</i>
OASIS	Organization for the Advancement of Structured Information Stan- dards, <i>Ein internationales Konsortium für die Entwicklung von Standards für Dokumenten-, Geschäfts- und Webtechnologien. – Webseite: http://www.oasis-open.org/.</i>

OCR	Optical Character Recognition, <i>Die automatische Erkennung von Text in Rasterbildern.</i>
OHCO	Ordered Hierarchy of Content Objects, <i>Ein Modell für Text als baum-strukturierte Schachtelung von Inhalten [DDMR90].</i>
OLAP	On-line Analytical Processing, <i>Zugriffsmodus auf Datenbanksysteme, bei dem das Nutzungsszenario des Auslesens und Auswertens großer Datenmengen im Vordergrund steht.</i>
OLTP	On-line Transaction Processing, <i>Zugriffsmodus auf Datenbanksysteme, bei dem ein schnelles Antwortverhalten bei der Prozessierung kleinerer bis mittelgroßer Datenmengen im Vordergrund steht.</i>
OSI	Open Systems Interconnection, <i>Eine Initiative der International Organization for Standardization (ISO) zur Standardisierung verschiedener Aspekte der Kommunikation in Computernetzwerken.</i>
OWL	OWL Web Ontology Language, <i>Eine vom WWW Consortium (W3C) standardisierte Beschreibungssprache für Ontologien [Version 1: MH04], [Version 2: OWL09a, OWL09b].</i>
PARTHENOS	Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies, <i>Ein europäischer Verbund aus Infrastrukturprojekten für die Geisteswissenschaften, der die Zusammenarbeit seiner Partnerorganisationen fördern und leiten soll. – Webseite: http://parthenos-project.eu/.</i>
PDF	Portable Document Format, <i>Ein Binärformat für Dokumente, welches hauptsächlich zur plattformunabhängigen Bildschirmanzeige oder für die Druckvorstufe verwendet wird [ISO32000].</i>
POS	Part-of-Speech, <i>Die Lexikalische Kategorie; Klassen von Wörtern, die gemeinsame morphosyntaktische Eigenschaften aufweisen.</i>
RDF	Resource Description Framework, <i>Mehrere Standards des W3C zur URL-basierten Identifizierung und Beschreibung von Entitäten und Metadaten im Kontext des Semantic Web.</i>

RDFS	RDF Schema, <i>Eine Sprache zur Schemadefinition für RDF-Daten, die grundlegende Funktionen zur Beschreibung von Ontologie enthält.</i>
REST	Representational State Transfer, <i>Eine Bezeichnung für Webservices, die bestimmten Kriterien, wie z. B. einem nach außen zustandslosen Verhalten genügt [Fie00].</i>
SGML	Standard Generalized Markup Language, <i>Eine textuelle, auf Tags basierende und hierarchische Auszeichnungssprache für Daten.</i>
SKOS	Simple Knowledge Organization System, <i>Ein für die Nutzung im Semantic Web entwickeltes Vokabular zur Wissensorganisation [MB09].</i>
SOAP	Simple Object Access Protocol, <i>Ein von der W3C entwickeltes und als "leichtgewichtig" bezeichnetes Protokoll für Informationsaustausch und den Betrieb von Webservices auf der Basis von XML. [BEK⁺04].</i>
SPARQL	SPARQL Protocol And RDF Query Language, <i>Eine deklarative Abfragesprache für graphförmige Daten, die nach dem RDF-Datenmodell modelliert wurden.</i>
SQL	Structured Query Language, <i>Eine seit den 1970er Jahren entwickelte Abfragesprache für relationale Datenbanksysteme [ISO9075].</i>
TaDiRAH	Taxonomy of Digital Research Activities in the Humanities, <i>Eine Initiative zur formellen Beschreibung von geisteswissenschaftlichen Forschungsaktivitäten.</i>
TCP	Transmission Control Protocol, <i>Ein verbindungsorientiertes, IP-Pakete vermittelndes Internet-Transportprotokoll mit Optionen zur Flusssteuerung.</i>
TEI	Text Encoding Initiative, <i>Eine Initiative für die Erarbeitung von Kodierungsrichtlinien und Austauschformaten für wissenschaftlich erfasste Textquellen [TEIP5]. – Webseite: http://www.tei-c.org/.</i>

TMQL	Topic Maps Query Language, <i>Eine Abfragesprache für Topic Maps, deren Standardisierung (als ISO 18048) nicht abgeschlossen wurde.</i>
UIMA	Unstructured Information Management Architecture, <i>Eine Programminfrastruktur für den Umgang mit unstrukturierten Daten und für automatische Sprachverarbeitung; ehemals von IBM entwickelt, mittlerweile von der Organization for the Advancement of Structured Information Standards (OASIS) standardisiert.</i>
URI	Uniform Ressource Identifier, <i>Eine Zeichenkette zur Identifikation von Ressourcen im Kontext des WWW [RFC1630].</i>
URL	Uniform Ressource Locator, <i>Eine Adresse (URI), die Ort und Zugriffsart einer digitalen Ressource (z. B. Webseite) angibt.</i>
URN	Uniform Ressource Name, <i>Eine Bezeichnung (URI), die Name und Namensschema einer digitalen Ressource angibt.</i>
UTF	Unicode Transformation Format, <i>Eine Familie verschiedener Binärkodierungen für Unicode-Zeichen.</i>
VIAF	Virtual International Authority File, <i>Eine weltweite Initiative für den Zusammenschluss von Normdatensammlungen. – Webseite: http://viaf.org/.</i>
W3C	WWW Consortium, <i>Ein internationales Gremium, welches Protokolle und Standards für ein stetiges Wachstum des WWW entwickelt. – Webseite: http://www.w3.org/.</i>
WSDL	Web Service Description Language bzw. Web Service Definition Language, <i>Eine vom W3C standardisierte Beschreibungssprache für die verschiedenen Schnittstellenaspekte von Webservices.</i>
WWW	World Wide Web, <i>Der Verbund verteilter und durch Hyperlinks verknüpfter Informationsressourcen im Internet, die als Service (hauptsächlich über das HTTP) bereitgestellt werden.</i>

XML	Extensible Markup Language, <i>Eine aus SGML hervorgegangene, textuelle und auf Tags basierende hierarchische Auszeichnungssprache für Daten, s. [BPSM⁺06].</i>
XSL	XML Stylesheet Language, <i>Eine Sammlung von XML-basierten Verfahren für die Umformung von XML-Inhalten.</i>
XSL-FO	XSL Formatting Objects, <i>Eine XML-basierte Sprache für die Repräsentation der Rohinhalte und (ausgabeorientierten) Formatierungen von Dokumenten.</i>
XSLT	XSL Transformation, <i>Eine XML-basierte Sprache für die programmatische Transformation von XML-Dokumenten in andere Document Object Model (DOM)-Repräsentationen.</i>

Abbildungsverzeichnis

2.1	Einordnung der e-Humanities ins Fächergefüge	19
3.1	Minimales Textdatenmodell	76
3.2	Schema der tokenzentrierten Datenmodellierung	77
3.3	Verknüpfungsvarianten für Hierarchie und Sequenz	78
3.4	Verknüpfung von Knoten für direkte Hierarchiesprünge	80
3.5	Architektur und Komponenten der Kadmos-Umgebung	82
3.6	Asynchrone Client-Server-Kommunikation	86
3.7	Auftretende Zeichen in Editionen altgriechischer Texte	92
3.8	Nutzerspezifische Metadaten	101
3.9	Zusammenfassung orthographischer Varianten in der Stichwortsuche . .	105
3.10	Verknüpfung von Konzepten, Vokabular und Dokumenten	107
3.11	Tag Pies von grammatikalischen Wortvarianten	111
3.12	Gewichtung für Kookkurrenzterme	112
3.13	Visualisierung aggregierter Konkordanzdaten	113
3.14	Abfragezeit nach Umgebungsgröße	116
3.15	Normalisierte Types nach Tokenanzahl	117
3.16	Abfragebeschleunigung durch parallele Verarbeitung	119
3.17	Visualisierung einer Abfrage für direkte Wortumgebungen	124
3.18	Übertragung und Darstellung von POS-Annotationen	128
4.1	Interaktive Visualisierung von Personennamen	138
4.2	Dokument-Personen-Netzwerk	139
4.3	„Ancient Word Explorer“	142
4.4	Detailansicht für Toponymkookkurrenz	143
4.5	Sequenz von Toponymnetzwerken	143
4.6	Netzwerkverbindungen und Kartenabbildung	144
4.7	Verknüpfung von Büchern und Autoren	154
4.8	Kartogramm von Herkunftsländern	155

4.9	Geschlechteranteil über die Zeit	156
4.10	Koautoren-Netzwerk	159
4.11	Interaktions-Schwellwerte für die Graphinduktion	163
4.12	Handlungsverlauf in Dramen	164
4.13	Übersetzungsvarianten	165
4.14	Beispiel für n-Gramm-Entropie	169
4.15	Beispiel für Blockentropie-basierte Maßzahl	171
4.16	QQ-Diagramm: Sequenzlänge gegen entropiebasierte Maßzahl	172
4.17	Zusammenhängende Lösungsgraphen	174
4.18	Wortlängenverteilung	175
4.19	Distanzberechnungen im Lösgraphen	176
4.20	Oberfläche von „Gremlin’s Property Graph Lab“	180
4.21	Ergebnistypen in „Gremlin’s Property Graph Lab“	181
B/1	Zeichen in altgriechischen Texten nach Häufigkeit	246
B/2	Ereignisbezogene Stichwortsuche	247
B/3	Verbundene Wortkontexte im „Kadmos Navigator“	248
B/4	Dewey-Klassifikationsnummern	249
B/5	Erweitertes Beispiel für Blockentropie-basierte Maßzahl	250

Quelltextverzeichnis

3.1	Schemadefinition in Kadmos	84
3.2	Import eines Dokuments über direkte Methodenaufrufe	88
3.3	Benchmarking-Code	117
3.4	Anlegen einer neuen API-Definition	123
3.5	Ausschnitt aus der Tagger-Wrapperklasse	127
C/1	Installationsschritte für Kadmos	251

Tabellenverzeichnis

3.1 Speicherbelegung und Abfrageaufwände	79
3.2 Nummerierungspräfixe und allgemeine Abschnittsbezeichner	90
3.3 Performance mit beschränktem Arbeitsspeicher	120
A Liste verwendeter Korpora	245

Anhang A

Verwendete Korpora

Korpus	Titel und Beschreibung	Frei verfügbar?
BTL	<i>Bibliotheca Teubneriana Latina</i> Digitalisierte Teubner-Editionen lateinischer Literatur von der Antike und Spätantike bis zum Neulatein, bezogen vom Verlag Walter de Gruyter im Rahmen des BMBF-geförderten Projektes „eXChange“ http://www.degruyter.com/view/db/btl	✗
COPPER-FIELD	„David Copperfield“ von Charles Dickens Einzeldokument-Korpus, bezogen vom Project Gutenberg, importiert ohne Präambel und Lizenztext http://www.gutenberg.org/files/766/766-0.txt	✓
DRAMEN	<i>Gemeinfreie deutschsprachige Dramen</i> 38 Dramen deutscher Autoren (u.a. von Arnim, Goethe, Grabbe, Lessing, von Kleist und Tieck), bezogen von der DVD „Gutenberg-DE Edition 13“ http://gutenbergshop.abc.de/catalog/index.php?cPath=29	✓
HAMLET	<i>The Shakespeare Quartos Archive</i> Sammlung von 32 Quarto-Ausgaben von Shakespeare's Hamlet im TEI-Format, Drucke zwischen 1603 und 1637 http://www.quartos.org/	✓

Fortsetzung auf der Folgeseite

Korpus	Titel und Beschreibung	Frei verfügbar?
HERODOT	Herodots „Historien“ (ἱστορίαι) Historisch-geographische Texte in neun Büchern im TEI-Format aus dem Perseus-Projekt, in Griechischer Version und in Englischer Übersetzung (von A. D. Godley), letztere mit NER-Tags incl. Geo-Kodierung http://www.perseus.tufts.edu/hopper/dltext?doc=Perseus:text:1999.01.0125 http://www.perseus.tufts.edu/hopper/dltext?doc=Perseus:text:1999.01.0126	✓
NYT	The New York Times Annotated Corpus Sammlung von über 1,8 Millionen Artikeln aus der New York Times, die zwischen dem 1. Januar 1987 und dem 19. Juni 2007 abgedruckt wurden. http://catalog.ldc.upenn.edu/LDC2008T19	✗
OTHELLO	Parallelstellen aus Othello-Übersetzungen 38 deutsche Übersetzungen und Theater-Bearbeitungen von Akt 1, Szene 3 aus Shakespeares „Othello“, zusammengestellt vom Projekt „Translation Arrays: Version Variation Visualization“ http://www.delightedbeauty.org/vvv	✓/ ✗
PAPYRI	Duke Databank of Documentary Papyri Umfangreiche Sammlung antiker Papyri-Texte in der Version „Epiduke“ des Kings College London http://epiduke.cch.kcl.ac.uk/	✓
REKTOREN	Leipziger Rektoratsreden 1871-1933 Antrittsreden der neu gewählten und Jahresberichte der scheidenden Rektoren anlässlich des feierlichen Rektoratswechsels, 1871–1933, [Häu09], bezogen vom Universitätsarchiv Leipzig http://www.degruyter.com/view/product/40375	✗

Fortsetzung auf der Folgeseite

Korpus	Titel und Beschreibung	Frei verfügbar?
REUTERS	Reuters-21578 Text Categorization Test Collection 21578 englischsprachige Agenturmeldungen zu Wirtschaftsthemen aus dem Jahr 1987, zur Verfügung gestellt von der Carnegie Group und Reuters http://www.daviddlewis.com/resources/testcollections/reuters21578/	✓
SHAKES- PEARE	Shakespeare 35 deutsche Versionen verschiedener Dramen Shakespeares, (in Übersetzung u. a. von Baudissin, Schiller, Schlegel, Tieck und Wieland), bezogen von der DVD „Gutenberg-DE Edition 13“ http://gutenbergshop.abc.de/catalog/index.php?cPath=29	✓
SHC	Shakespeare His Contemporaries 510 englischsprachige Dramen im TEI-Format aus der Zeit von 1552 bis 1662 http://shakespearehiscontemporaries.northwestern.edu/	✓
STASI	Die DDR Im Blick der Stasi Sammlung der geheimen Stimmungs- und Lageberichte der „Zentralen Auswertungs- und Informationsgruppe“ des Ministeriums für Staatssicherheit an die Staatsführung der DDR, umfasst die bisher editorisch abgedeckten Jahrgänge 1953, 1961, 1976, 1977 und 1988 http://www.ddr-im-blick.de/	✗
TLG	Thesaurus Linguae Graecae Umfangreiche Sammlung griechischer Texte aus der klassischen Periode vom 8. vorchristlichen bis 6. Jahrhundert unserer Zeit sowie aus der anschließenden byzantinischen Zeit, bis 1453; insgesamt ca. 76 Mio. Token umfassend. Die Nutzung erfolgte über das BMBF-geförderte Projekt „eXChange“ http://stephanus.tlg.uci.edu/	✗

Fortsetzung auf der Folgeseite

Korpus	Titel und Beschreibung	Frei verfügbar?
VOYNICH	Voynich-Manuskript Einzelndokument-Korpus, das verschiedene Transkriptionen eines mehr als einhundert Seiten umfassenden, wahrscheinlich aus dem frühen 15. Jahrhundert stammenden und in einer Geheimschrift verfassten Werkes bündelt. http://www.ic.unicamp.br/~stolfi/voynich/Notes/060/L16+H-eva/text16e7.evt	✓

Ergänzende Grafiken

Abbildung B/1: Auftretende Unicode-Zeichen in digitalen Editionen altgriechischer Texte, sortiert absteigend nach Häufigkeit

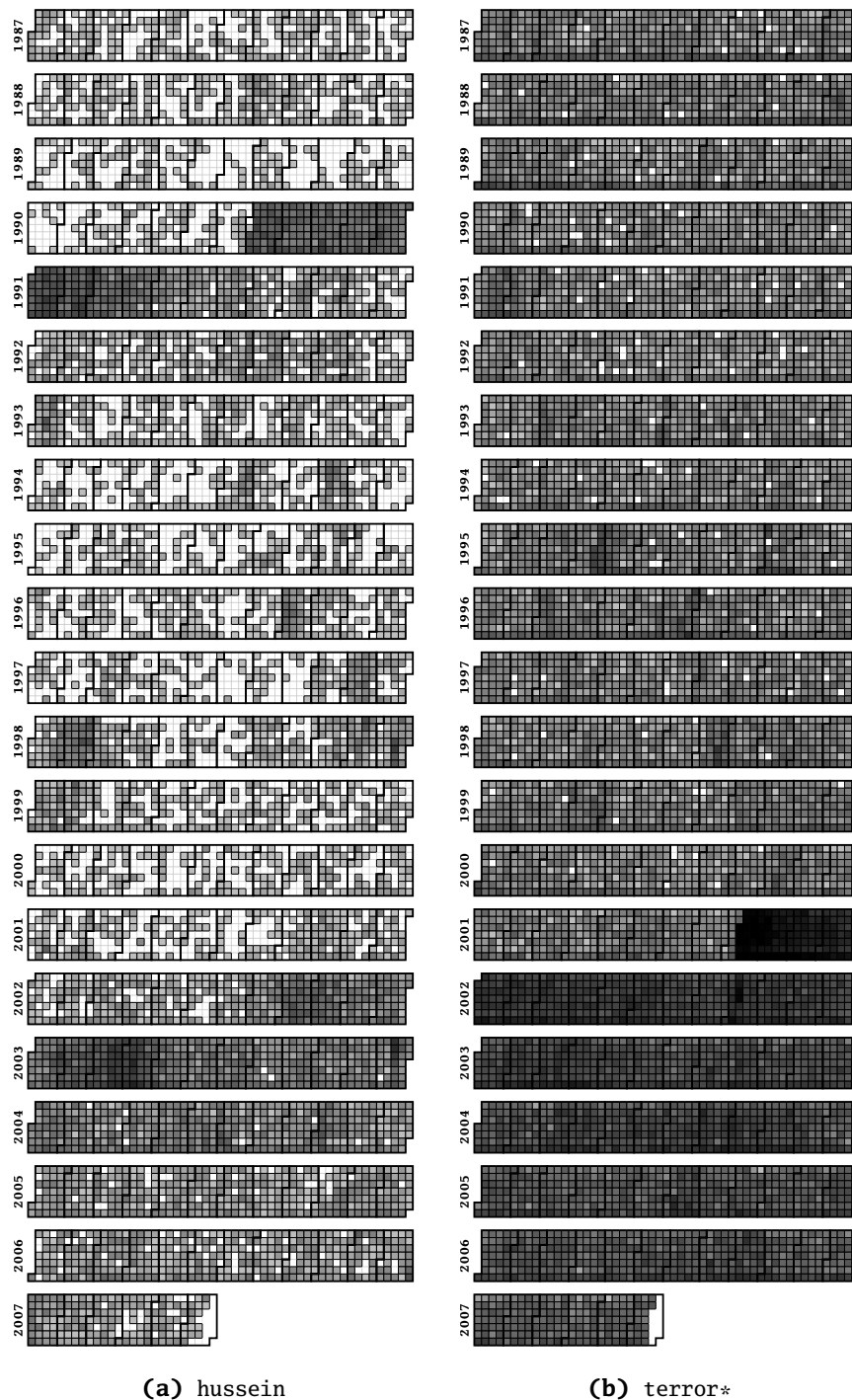


Abbildung B/2: Jahreskalender-Histogramme für die ereignisbezogene Stichwortsuche, (a) dominiert vom zweiten und dritten Golfkrieg und (b) in Form vieler Einzelereignisse, überschattet von Berichten zu den New Yorker Anschlägen vom 11. September 2001.

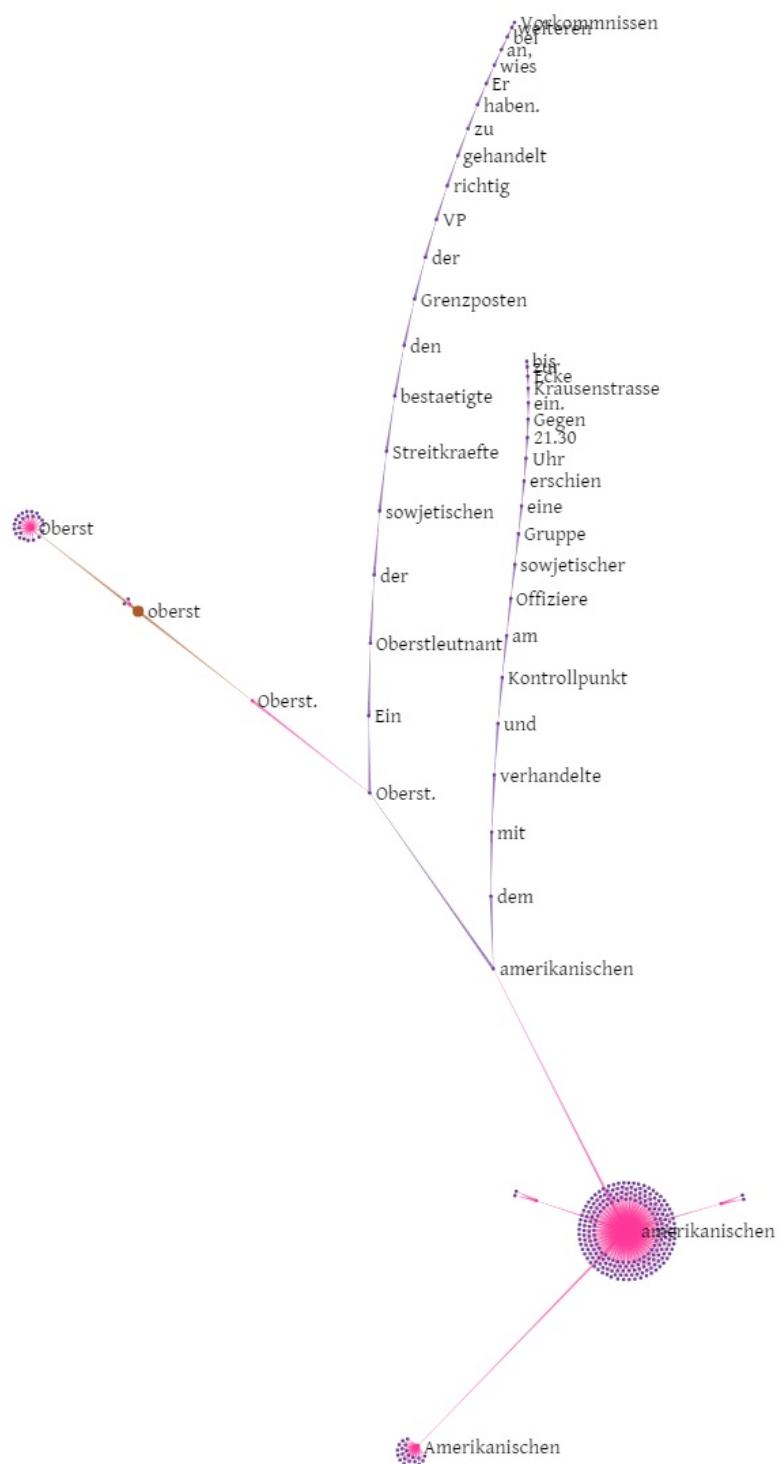


Abbildung B/3: Verbundene Wortkontexte für die normalisierten Types **amerikanischen** und **oberst** aus dem **STASI-Korpus**, visualisiert im „Kadmos Navigator“ (Token tragen dabei die Labels ihrer Types)

249

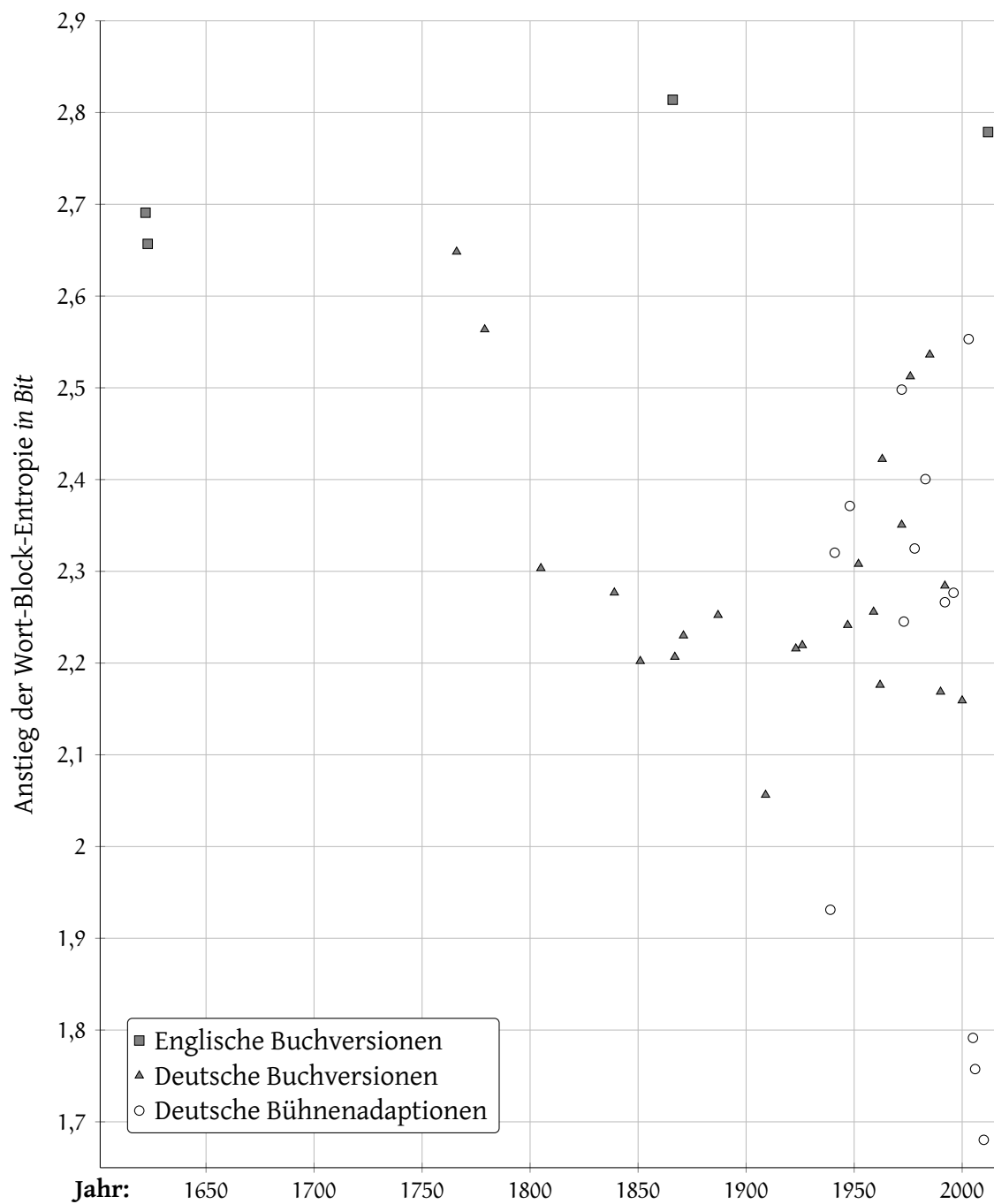


Abbildung B/5: Blockentropie-basierte Maßzahl für alle Varianten aus dem [OTHELLO-Korpus](#)

Anhang C

Anleitung zur Inbetriebnahme der Kadmos-Umgebung

Der aktuelle Arbeitsstand der Kadmos-Kernumgebung wird gemeinsam mit ausgewählten Plugins unter der Adresse <http://kadmos.textgraph.science/> veröffentlicht, präsentiert und dokumentiert. Für die langfristige Weiterentwicklung von Kadmos ist geplant, die Softwareentwicklung aus dem bisherigen isolierten Projektumfeld herauszulösen und in einen stetigen und offenen Entwicklungsprozess zu überführen. Diese Arbeiten werden künftig auch über die Kadmos-Webseite koordiniert. Ziel ist es, eine kontinuierliche Weiterentwicklung entsprechend fachlicher Anforderungen aus den e-Humanities und technologischer Erkenntnisse aus der Informatik zu erreichen.

Angesichts dessen ist es nicht angedacht, einen bestimmten Entwicklungsstand festzuhalten und als Teil dieser Arbeit anzusehen. Eine jeweils aktuelle Systemversion als Kombination stabiler Basisfunktionen und eventueller gerade in Ausarbeitung befindlicher experimenteller Funktionalität kann wie folgt bezogen werden:

```
1 apt-get install git jruby
2 jruby -S gem install bundler
3 git clone git://git.textgraph.science/kadmos
4 cd kadmos
5 jruby -S bundle install
6 jruby -J-Xmx8G -J-Dfile.encoding=UTF-8 lib/kadmos.rb
```

Quelltext C/1: Installationsschritte für Kadmos in typischer Debian-basierter GNU/Linux-Umgebung und Startbefehl mit 8 GB maximal zugewiesenem Arbeitsspeicher

Auf die Installation des Versionsverwaltungssystems Git und der Laufzeitumgebung JRuby folgt die Installation des Ruby-Pakets Bundler, über das die Laufzeitabhängigkeiten von Kadmos verwaltet werden. Nach dem Herunterladen des Quelltextes werden diese Abhängigkeiten aufgelöst und die benötigten Programmbibliotheken geladen. Anschließend kann Kadmos gestartet werden.

Nun sollte die korrekte Einrichtung und Ausführung des Systems überprüft werden. Beim Aufruf von http://localhost/api/system_status wird vom Server periodisch eine JSON-kodierte Statusmeldung mit aktuellen technischen Details der Laufzeitumgebung entsprechend dem folgenden Beispiel gesendet: ¹

```
1 { "process_memory":6364860, "databases_disk_space": [{ "btl":18244424 }, { "  
  ⌵ copperfield":364268 }, { "default":107042260 }, { "reuters":2683408 }, { "stasi "  
  ⌵ :1606812 }, { "system":444 }, { "tlg":107026384 }, { "voynich":50936 } ], "jvm_uptime "  
  ⌵ :1999595, "jvm_heap_memory":4060340, "jvm_non_heap_memory":92996, "  
  ⌵ jvm_recent_system_load":1.08, "jvm_thread_count":166, "  
  ⌵ jvm_currently_loaded_classes":15481, "jvm_total_loaded_classes":15879 }
```

Wird in der Adresse des Aufrufs „api“ durch „rest-api“ ersetzt, erfolgt die Rückmeldung einmalig.

Zum Anlegen einer neuen Korpus-Datenbank muss nur (z. B. mit dem mit dem Kommandozeilenbefehl „mkdir ./db/Datenbankname“) ein neues Verzeichnis im db-Unterverzeichnis der Kadmos-Anwendung erstellt werden. Beim anschließenden Aufruf der Adresse <http://localhost/rest-api/corpora?update=true> wird in dieses neue Verzeichnis eine leere lokale Datenbank (Berkeley-DB-Backend) mit entsprechenden Indexstrukturen (Lucene-Index) initialisiert. Standardmäßig wird in Kadmos das Korpus mit dem Namen „default“ verwendet. Sollen sich die folgenden Abfragen auf einen alternativen Datenbanknamen beziehen, so ist dieser als HTTP-Parameter z. B. mit `&corpus=Datenbankname` anzufügen.

Für diese neue Korpus-Datenbank oder für die default-Datenbank kann nun der Importvorgang eines Beispielkorpus initiiert werden. Durch den Aufruf der Adresse <http://localhost/api/import?importer=demo> wird ein minimales Einzeldokument-Korpus geladen, welches nur den Beispieltext aus Abbildung 3.1 auf Seite 76 enthält. Sollte ein größeres Test-Korpus benötigt werden, muss entsprechend ein anderer Importer verwendet werden. Durch Eingabe von `importer=reuters` wird z. B. das [REUTERS-Korpus](#) aus dem Internet geladen und in die Graphdatenbank übernommen.

¹Zahlen zur Speichernutzung sind dabei in Kilobyte angegeben.

Sobald die Korpora übernommen sind, liefern die textstatistischen [API](#)-Endpunkte vollständige Antworten zurück. Die 100 häufigsten Types des Korpus lassen sich über http://localhost/rest-api/ranked_types ermitteln. So z. B. für den Demo-Import:

```
1 { "leaves":2, "the":2, "The":2, "tonight.":1, "ship":1, "brown. ":1, "are":1, "All":1, "  
  ↪ dog. ":1, "lazy":1, "over":1, "jumps":1, "fox":1, "brown":1, "quick":1}
```

Ein Aufruf von http://localhost/rest-api/ranked_normalized_types liefert die selbe Statistik für normalisierte Types:

```
1 { "the":4, "leaves":2, "brown":2, "tonight":1, "ship":1, "are":1, "all":1, "dog":1, "  
  ↪ lazy":1, "over":1, "jumps":1, "fox":1, "quick":1}
```

Diese und weitere „erste Schritte“ nach der Einrichtung des Systems sowie weitere beispielhafte Anwendungsfälle werden in geeigneter Form unter <http://kadmos.textgraph.science/> dokumentiert und, wie oben beschrieben, jeweils um aktuelle Informationen ergänzt.



Thomas Efer, 2016–2017

Typesetting

MikTeX 2.9
KOMA-Script 3.15
Babel 3.9l
pdfTeX 3.1415926-2.6-1.40.15

Typefaces

Gentium by Victor Gaultney
LuxiMono by Kris Holmes & Charles Bigelow
■ by Dave Gandy

GRAPHDATENBANKEN FÜR DIE TEXTORIENTIERTEN E-HUMANITIES

In diesem Buch wird ein Beitrag der Informatik zum interdisziplinären Feld der *e-Humanities* beschrieben, welcher darauf abzielt, die Palette digitaler Methoden in der textorientierten geistes- und sozialwissenschaftlichen Forschung zu erweitern. Es wird exemplarisch untersucht, inwieweit alternative Methoden der digitalen Textrepräsentation als Grundlage für innovative Rechtersysteme genutzt werden können. Die inhaltlichen Aspekte reichen dabei von der Korpusverwaltung über Textmodellierung und -annotation bis hin zur durch *Text-Mining* unterstützten Korpusauswertung. Für diese Aspekte wird eine Lösung unter Nutzung aktueller Graphdatenbanksysteme vorgestellt.

Das Buch bietet dabei nicht nur einen anwendungsorientierten wissenschaftlichen Überblick über bestehende Technologien, aktuelle Probleme und Lösungsansätze, sondern stellt mit dem Text-Rechtersystem „*Kadmos*“ auch praktische Arbeiten und Softwarekomponenten vor, deren Einsatzmöglichkeiten anhand ausgewählter Anwendungsfälle aus der *e-Humanities*-Forschung illustriert werden.

Über den Autor

Thomas Efer ist promovierter Informatiker. Er forscht an der Universität Leipzig, wo er seit 2011 technische und methodische Fragestellungen in zahlreichen Projekten – insbesondere aus dem Bereich der *e-Humanities* – bearbeitet hat. Seine Forschungsschwerpunkte liegen in der interdisziplinären Konzeption und Anwendung von Verfahren der automatischen Sprachverarbeitung und graphbasierten Wissensmodellierung sowie in der Weiterentwicklung von *Big-Data*-Technologien.

 Open Access 



<https://e-humanities.textgraph.science/>