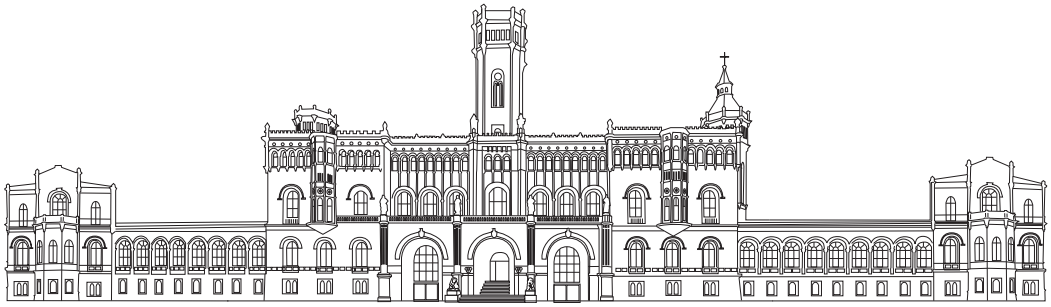


Der Hannover Concordancer und das Hannover Advanced Academic Writing Corpus: Eine korpuslinguistische Software mit dem dazugehörigen Dissertationskorpus für den Einsatz in Schreibberatungen

Von der Philosophischen Fakultät
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades eines Doktors der Philosophie (Dr. phil.)
genehmigte Dissertation

von Tobias Gärtner, M.Ed.

Erscheinungsjahr 2023

Referent: Prof. Dr. Peter Schlobinski

Korreferent: Prof. Dr. Ulrich Heid

Tag der Promotion: 15. November 2023

„Denn unser Wissen ist Stückwerk”

1. Brief an die Korinther 13:9α

Für Friedrich und Ludwig Gärtner

In tiefer Dankbarkeit

Peter Schlobinski Ulrich Heid
Christina Gärtner Katja Gärtner
Konrad Schäfer Tim Botzkowski
Katharina & Fabian Lochmann

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Internetquellenverzeichnis

Abkürzungsverzeichnis

ANOVA Analyse der Varianz. 170

ARF Average Reduced Frequency. V, 91–93

ARRF Attribute-Relation File Format. 64

AWS Amazon Web Services. 131

BNC British National Corpus. V, 79, 94–97, 100, 110, 138, 185

CHI Calinski-Harabasz Index. 182, 184

CIA Contrastive Interlanguage Analysis. 170

COCA Corpus of Contemporary American English. 52, 79, 95–97, 100

CP Credit Points. IV, VII, VIII, 33–40, 210

CSS Cascading Style Sheets. 77, 102, 104, 124, 128, 130

CTM Correlated Topics Model. 182

CWB Open Corpus Workbench. II, 79–81

DH Duale Hochschule. 44

DPM Dirichlet Process Mixture Model. IV, 38–40

DTM Dokumenten-Term Matrix. 155, 187

DWDS Digitales Wörterbuch der deutschen Sprache. 52

ECTS European-Credit-Transfer System. 33

ETS Educational Testing Service. 21, 184, 185

FArc Fakultät für Architektur und Landschaft. V, VIII, 34, 36, 57–59, 63, 209, 210, 213, 214

FBau Fakultät für Bauingenieurwesen und Geodäsie. V–VIII, 32, 34, 36, 57, 59–61, 63, 67, 70–73, 132, 161–168, 209, 210, 215

FElt Fakultät für Elektrotechnik und Informatik. V–VIII, 28, 32, 34, 36, 41, 42, 57, 59–61, 63, 67, 70, 71, 73, 74, 87, 132, 149, 150, 152, 161–168, 174, 175, 178, 179, 209, 210, 216

FH Fachhochschule. 44

Fjur Juristische Fakultät. 57

FMas Fakultät für Maschinenbau. V–VIII, 28, 32, 34, 36, 42, 57, 59–61, 63, 67, 70, 71, 73, 75, 132, 161–168, 209, 210, 217

FMat Fakultät für Mathematik und Physik. V, VIII, 34, 57, 59, 63, 67, 69, 70, 132, 209, 210, 218, 219

FNat Naturwissenschaftliche Fakultät. V, VIII, 34, 41, 42, 57, 59, 63, 67, 69, 70, 132, 209, 210, 220, 221

FPhi Philosophische Fakultät. IX, 28, 34, 36, 42, 57, 58, 63, 67, 132, 209, 210, 222, 223

FWir Wirtschaftswissenschaftliche Fakultät. IX, 34, 57, 63, 67, 70, 132, 209, 210, 224, 225

GbR Gesellschaft bürgerlichen Rechts. 23

GDEX Good Dictionary Examples. 89

GLM Generalised Linear Models. 172

GP Geographic Profiling. 38

GUI Graphical User Interface. 31, 128

HAAWC Hannover Advanced Academic Writing Corpus. 57, 59, 188, 189

HanConc Hannover Concordancer. II–V, VII, X, 2, 4, 6, 16–19, 21, 31, 41–44, 52, 54–57, 60, 77–79, 86, 87, 98, 101–112, 117, 118, 123, 126–140, 142, 144–149, 151, 153, 154, 156, 157, 160, 165, 167–173, 176–180, 182, 185–190

HTML HyperText Markup Language. X, 20, 21, 77, 83, 102, 104, 123–126, 128, 130, 136, 169

IDE Integrated Development Environment. 129

KIQS Konzepte und Ideen für Qualität im Studium. 78

KVV Kommentiertes Vorlesungsverzeichnis. 34

KWIC Key Words in Context. VII, 52, 76, 80, 81, 84, 86–89, 97, 99, 100, 104, 110, 126, 130, 133–139, 146, 152, 168, 176, 179–181, 184, 187

LDA Latent Dirichlet Allocation. 182

LSA Latent Semantic Analysis. VIII, 52, 126, 127, 154–157, 160, 161, 165–168, 181

LSI Latent Semantic Indexing. 155

LUH Leibniz Universität Hannover. II, IV, 2, 3, 7, 8, 10, 17, 21, 24, 26–28, 31, 33, 34, 39, 41, 44, 50, 55, 57, 59, 75, 78, 88, 104, 107, 148, 177, 186, 189

MI Mutual Information. III, V, 82, 84, 138–140, 143–145, 148, 176–178

MTLD Measure of Textual Lexical Diversity. 69

NER Named Entity Recognition. 86, 117, 134

NLP Natural Language Processing. 112

PCA Principal Component Analysis. 157

PH Pädagogische Hochschule. 44

PoS Part-of-Speech. 86, 88, 101, 105, 111–113, 117, 130, 133–135, 139, 140, 142, 145

RAM Random-Access Memory. 17, 77, 127, 129, 132, 145, 160

RegEx Reguläre Ausdrücke/Regular Expressions. VII, 80, 86, 88, 99, 100, 110, 120, 133

SPbSPU Polytechnische Staatliche Universität St. Petersburg. 78

STTS Stuttgart-Tübingen-Tagset. 105, 113

SVD Singular Value Decomposition. VII, VIII, 155, 157–159, 181, 182

TDM Term-Dokumenten Matrix. VII, 20, 64, 127, 154–157, 160, 168, 180–182

TF-IDF Term Frequency-Inverse Document Frequency. 62, 127, 156, 181, 182

TIB/UB Technische Informationsbibliothek und Universitätsbibliothek. 57, 78, 107

TOEFL Test of English as a Foreign Language. 21, 184

TTR Type-Token Ratio. II, 67, 69–71, 82

WST WordSmith Tools. II, 81–83, 85, 87, 89, 92, 97, 99, 135

XML Extensible Markup Language. II, V, 80, 83, 95, 110–112, 117

Kapitel 1

Einleitung

Bildungseinrichtungen funktionieren, trotz aller romantischen Bildungsideale, nach wirtschaftlichen Prinzipien. Eine höhere Bildungseinrichtung wie eine Universität oder Fachhochschule wird in Fakultäten unterteilt und diese in Institute. An diesen Instituten arbeiten Professor_innen[1] und ein akademischer Mittelbau. Zusätzlich unterstützen Mitarbeiter_innen den Lehrbetrieb in Technik und Verwaltung. Sowohl das Personal als auch die Infrastruktur kosten Geld. Das bedeutet, dass mit begrenzten finanziellen Mitteln eine gewisse Anzahl an Studienplätzen realisiert werden kann. Da vielfach die Anzahl an Studieninteressierten größer ist als die Anzahl an Studienplätzen, werden nur die erfolgversprechendsten Bewerber_innen akzeptiert. Manche Studiengänge nutzen die ersten Semester, um die Anzahl der Studierenden weiter zu reduzieren. Im Verlauf des Studiums versucht die Bildungseinrichtung durch weitere Angebote wie Schreibzentren, Studierende zu ihrem Abschluss zu führen, um die aufgewendeten Ressourcen sinnvoll eingesetzt zu haben. Auch wenn diese Beschreibung überspitzt und gleichzeitig simplifiziert ist, so entspricht sie doch in Grundzügen einer wirtschaftlichen Betrachtung von höherer Bildung.

Viele Studiengänge befassen sich in ihren ersten Semestern vor allem mit den für das Fachgebiet notwendigen Grundlagen. Akademisches Schreiben gehört außerhalb der Geisteswissenschaften häufig nicht dazu. So kommt es vor, dass vor allem in ingenieur- und naturwissenschaftlichen Studiengängen erst in den letzten beiden Semestern wissenschaftliche Aufsätze geschrieben werden müssen. Um Komplikationen beim wissenschaftlichen Schreiben von Abschlussarbeiten zu vermeiden und damit die von den Bildungseinrichtungen eingesetzten Mittel zu bewahren, haben viele größere Universitäten und Fachhochschulen Schreibzentren eingerichtet. Diese Schreibzentren sollen Studierende in ihren Schreibprozessen begleiten und damit die Wahrscheinlichkeit einer erfolgreichen wissenschaftlichen Arbeit erhöhen.

Schreibzentren stehen vor der Herausforderung, dass das Personal, welches sie rekrutieren, vor allem aus Geisteswissenschaftler_innen besteht. Diese kennen sich zwar mit dem Schreiben im Allgemeinen und im Speziellen mit dem Schreiben in geisteswissenschaftlichen Disziplinen aus, sind jedoch häufig mit Studierenden aus anderen akademischen Traditionen konfrontiert.

[1] Diese Arbeit verwendet für Personengruppen zur besseren Lesbarkeit die weiblichen Artikel und Adjektive. Um alle Geschlechter zu berücksichtigen, wird wird entweder ein Gerundium verwendet oder vor die weibliche Wortendung „in" ein Unterstrich „_" eingefügt.

Vor allem, wenn die Studierenden ihre Arbeiten in einer Fremdsprache verfassen müssen, ergeben sich zusätzliche Herausforderungen. In dieser Situation reichen allgemeine Hinweise zu Herangehensweisen an akademisches Schreiben und die Vermittlung von unterschiedlichen Zitierweisen nicht aus.

Wenn innerhalb der Schreibberatung auf einzelne Absätze und Formulierungen in Abschlussarbeiten eingegangen werden soll, stellt sich die Frage, mit welchen Mitteln die Erfahrungslücken zwischen den akademischen Traditionen der Schreibberater_innen und der Studierenden zu überbrücken sind. Zwar gibt es für einzelne akademische Disziplinen fachspezifische Handbücher, Nachschlagewerke und Wörterbücher, jedoch sind diese in der Breite unzureichend vorhanden und in der Tiefe teils unbrauchbar, um Schreibberater_innen dabei zu unterstützen, sich sprachlich in die Thematik ihrer Studierenden einzudenken. Studierende wissen sehr wohl, was die von ihnen verwendeten Fachbegriffe bedeuten. Es fehlen ihnen oftmals jedoch die Mittel, diese sinnvoll in den Kontext eines akademischen Aufsatzes einzubetten.

Korpuslinguistik würde hier Abhilfe schaffen. Mittels Korpora und einer entsprechenden Software ließe sich die Verwendung von Begriffen in ihrem Kontext betrachten und danach in eigenen Texten nachahmen. Wenn die Software und die Kenntnisse der Nutzer_innen es hergeben, ließe sich die Verwendung eines Suchbegriffs quantifizieren und damit feststellen, ob die einzelne Verwendung eine Ausnahme oder doch die Regel ist. Allerdings sind bisherige Werkzeuge vor allem auf die Bedürfnisse von Linguist_innen ausgerichtet oder aber basieren auf Textgrundlagen, die zu allgemein für Schreibberatungen sind. Aus diesem Grund haben sich korpuslinguistische Werkzeuge wie AntConc oder WordSmithTools bisher vielfach nicht durchgesetzt.

Die dieser Arbeit zu Grunde liegende Software soll eine Unterstützung für Schreibberatungen sein, um den Mangel an adäquaten Hilfsmitteln zu reduzieren. Mit Hannover Concordancer (HanConc) soll es Schreibberater_innen ermöglicht werden, fachspezifische Textsammlungen anzulegen, zu durchsuchen und die Ergebnisse adressatengerecht aufzubereiten. Damit wird die Beratung über wissenschaftliche Disziplinen hinweg vereinfacht und somit die Erfolgswahrscheinlichkeit von Schreibberatung für Studierende erhöht.

Die Arbeit ist wie folgt gegliedert: Zunächst wird der inhaltliche Rahmen abgesteckt. Anschließend wird die Zusammenarbeit von Schreibberater_innen und Studierenden innerhalb des akademischen Schreibprozesses analysiert. Am Beispiel der Leibniz Universität Hannover (LUH) werden die universitären Anforderungen an Schreibberatung ermittelt und mit dem tatsächlichen Aufwand eines Schreibberaters, d.h. in diesem Fall des Autors, verglichen. Mit Hilfe einer Umfrage wird überprüft, ob sich die Ergebnisse des Vergleichs mit den Erfahrungen anderer Schreibzentren an anderen deutschen Bildungseinrichtungen decken. Um die Anforderungen von Schreibberatungen an HanConc zu erfüllen, wurde ein Korpus bestehend aus allen Dissertationen der LUH erstellt. Dieses Korpus wird eingehend beschrieben und auf die Homogenität innerhalb und Heterogenität außerhalb der Fakultätsgrenzen überprüft. Abschließend wird HanConc im Vergleich zu bestehenden Werkzeugen als Antwort auf die skizzierten Anforderungen vorgestellt. Der Quellcode inklusive ausführlicher Dokumentation befindet sich in

einem gesonderten Repositorium.

1.1 Motivation

Schreibzentren sind trotz ihrer wissenschaftlichen Forschung Serviceeinrichtungen. Ziel dieser Einrichtungen ist die Betreuung von Studierenden beim Erstellen von schriftlichen Arbeiten. Es stellt sich die Frage, wie der Erfolg einer solchen Einrichtung zu bewerten ist. Im Gegensatz zur Qualität von wissenschaftlicher Forschung, die sich kaum quantifizieren lässt, kann der Erfolg eines Schreibzentrums durchaus gemessen werden. Dies ist auch notwendig, da die Ziele einer solchen Serviceeinrichtung darin liegen, die Zahl an erfolgreich bestandenen schriftlichen Leistungen zu erhöhen. Abgebrochene oder endgültig nicht bestandene Studien binden universitäre Kapazitäten und gehen vielfach mit persönlichen Konsequenzen für die Studierenden einher. Daher sollte ein Schreibzentrum als Ganzes und jede Schreibberatung als Teil dessen auf ihre Effizienz geprüft und gegebenenfalls angepasst werden.

Der Prozess des Schreibens kann auf vielfältige Weise unterstützt werden. Viele Schreibzentren bieten Kurse, Workshops in Kleingruppen, Schreibwerkstätten, Events und auch persönliche Schreibberatung an. Je größer und heterogener die zu betreuende Gruppe ist, desto oberflächlicher und allgemeiner fällt die Betreuung aus. Insbesondere fortgeschrittene Schreibende benötigen jedoch eine intensivere Unterstützung.

Die vorliegende Arbeit zielt daher vor allem auf persönliche Schreibberatung ab, wie sie auch das Multilinguale Schreibzentrum der LUH anbietet. Dabei befassen sich die Schreibberater_innen über einen längeren Zeitraum mit der Arbeit einzelner Studierender. Da die Schreibberatung allen Studierenden offen steht und die zu betreuenden Master- und Doktorarbeiten bereits hochgradig spezialisiert sind, werden die Schreibberater_innen häufig mit Themen konfrontiert, die von ihrer eigenen akademischen Ausbildung abweichen. Für das Schreibzentrum der LUH kommt für die Schreibberater_innen noch erschwerend hinzu, dass nur Studierende betreut werden, die nicht in ihrer Erstsprache schreiben.

Die vorliegende Arbeit setzt an oben beschriebenem Grundproblem an: Studierende und Schreibberater_innen haben unterschiedliche akademische Traditionen und können sich daher inhaltlich über das Geschriebene nicht austauschen; gleichzeitig ergibt sich aus dem Schreiben in einer Fremdsprache die Notwendigkeit einer sprachlich tiefgründigeren Betreuung. Bereits bestehende Hilfen wie Wörterbücher oder Internetsuchen sind vielfach unzureichend, da sie auf den Inhalt des Gesuchten und weniger auf die sprachliche Verwendung eingehen. Als potentielle Lösung werden Korpora mit entsprechender Software eingesetzt, um nach der sprachlichen Umgebung eines Begriffes zu suchen. Diese Korpussoftware erhöht die Komplexität allerdings noch um eine fachwissenschaftssprachliche Komponente. Die bisherigen Lösungen richten sich an Sprachwissenschaftler_innen oder Übersetzer_innen und weniger an Schreibberater_innen. Somit müssen sich die Schreibberater_innen in linguistische Konzepte und Fachvokabular einarbeiten, um diese Werkzeuge nutzen zu können. Mit dieser Arbeit wird Schreibberater_innen ein effektives und gleichzeitig einfach zu bedienendes Suchwerkzeug für die Nutzung von Kor-

pora angeboten.

1.2 Abgrenzung

Bei dem untersuchten Gegenstand dieser Arbeit handelt es sich um Software, die Schreibberater_innen bei ihrer Arbeit mit Korpora unterstützen soll. Ein Erfolg dieser Software ist gegeben, wenn sie in Schreibberatungen sinnvoll eingesetzt werden kann. Daher zielt diese Arbeit nicht darauf ab, ein Schreibberatungsmodell oder ein pädagogisches Konzept zu entwickeln oder den Einsatz von Korpora in Schreibberatungen zu beschreiben. Stattdessen soll ein einfach zu manipulierendes Werkzeug vorgestellt werden, das es Schreibberater_innen und Studierenden ermöglichen soll, ihre linguistischen fachwissenschaftssprachlichen Problemstellungen mit ausgewählten Korpora zu lösen, ohne sich selbst tiefgründig mit Linguistik auskennen zu müssen.

Auf Grund der Abgrenzung wird kein eigenes Schreibmodell entworfen. Statt eines eigenen Modells entwickelt Kapitel 2 eine systematische Beschreibung des Prozesses von Schreibberatungen. Welche pädagogischen und didaktischen Werkzeuge und Methoden in den Schreibberatungen eingesetzt werden, bleibt dabei unberührt.

1.3 Annahmen über Schreibzentren und Korpora

Einige Annahmen sind grundlegend für die Entwicklung des Hannover Concordancers (HanConc). Kommt bei Schreibberatungen ein Schreibmodell entgegen der hier getroffenen Annahmen zum Einsatz, so kann dies dazu führen, dass der Einsatz von Korpora und damit HanConc nicht zielführend ist. So kann eine inhaltliche Diskussion zwischen der Studierenden und der betreuenden Person über die zu schreibende Arbeit auch als Schreibberatung gelten, wenn zwar Hinweise zum Verfassen der Arbeit gegeben werden, jedoch nicht auf die sprachliche Ebene eingegangen wird. Hier auf den Einsatz von Korpora zu hoffen oder diesen vorzuschlagen erscheint wenig erfolgversprechend und wird daher verworfen. Die folgenden Axiome werden daher als grundlegend angenommen:

1. Es besteht eine zwangsläufige kommunikative Asymmetrie zwischen Schreibberater_innen und Studierenden

   Unabhängig davon, welchem Fachgebiet sich Schreibberater_innen und Studierende zugehörig fühlen, kann keine inhaltlich und methodisch symmetrische Kommunikation hergestellt werden. Selbst innerhalb eines Fachgebietes kann eine vollständige Kongruenz inhaltlicher Konzepte und Ideen nicht erreicht werden. Auch bei einer großen Schnittmenge bleiben viele Divergenzen bestehen (siehe Kapitel 3.1). Diese Divergenzen führen dazu, dass eine sprachliche Einbettung von Fachbegriffen als Hilfestellung nicht mehr geleistet werden kann und auf externe Hilfsmittel zurückgegriffen werden muss.

2. Schreibberatung von Studierenden, die nicht in ihrer Erstsprache schreiben, ist auch immer eine sprachliche Hilfestellung

   Sowohl die Literatur zu Schreibberatungen als auch viele Internetseiten einzelner Schreibzentren weisen darauf hin, dass die Struktur eines Textes und der Schreibprozess im Vordergrund einer Beratung stehen. Grammatik, Rechtschreibung, Morphologie und Kollokationen etwa sollen vielfach nicht explizit bearbeitet werden. Allerdings brauchen vor allem Studierende, die in einer anderen als ihrer Muttersprache schreiben, auch eine sprachliche Hilfestellung. Wie diese Hilfestellung tatsächlich ausfallen kann, soll nicht Thema dieser Arbeit sein. Es soll jedoch davon ausgegangen werden, dass diese Studierenden ein Interesse an der sprachlichen Verbesserung ihrer Arbeit haben und diese auch im Rahmen einer Schreibberatung erfolgen soll.

3. Es gibt einen Unterschied zwischen allgemeiner Wissenschaftssprache und Fachwissenschaftssprache

   Die Wissenschaftssprache der einzelnen Fachgebiete variiert nicht nur in Bezug auf Fachbegriffe. Zusätzlich müssen noch die Standards einer allgemeinen Wissenschaftlichkeit eingehalten und die weiteren Eigenheiten des jeweiligen Fachgebiets berücksichtigt werden. Schreibberater_innen können sich auf Grund der Vielfältigkeit der Fachgebiete, zu denen sie beraten sollen, entweder auf Fachgebiete spezialisieren oder sich auf eine allgemeinere Wissenschaftssprache zurückziehen. Vor dem Hintergrund der deutlich größeren Menge an Fachgebieten im Verhältnis zur Anzahl an Berater_innen pro Schreibzentrum ist davon auszugehen, dass eher Letzteres zutrifft. Sollten die Studierenden gerade aber mit der Fachwissenschaftssprache ihres Fachgebietes Probleme haben, so muss auch diese Lücke gefüllt werden.

   Meist können Studierende die Fachkonzepte, die in der zu schreibenden Arbeit vorkommen, inhaltlich richtig anwenden, jedoch fällt ihnen die sprachliche Einbettung schwer. Selbst die perfekte inhaltliche Durchdringung eines Konzeptes garantiert nicht, dass die entsprechend kollokierten Adjektive und Verben bekannt sind. Außerdem können englischsprachige Lehrveranstaltungen dazu führen, dass, wenn die Arbeit auf deutsch verfasst werden muss, die entsprechenden Fachbegriffe auf deutsch verwendet werden müssen. Die Schreibberater_innen, welche die Zielsprache zwar vielfach auf muttersprachlichem Niveau beherrschen, kennen jedoch meist die Fachkonzepte nicht (siehe Punkt 1) und können daher auch nicht unterstützen.

4. Die verwendete Fachwissenschaftssprache wird nicht ausreichend in Wörterbüchern erklärt

   Bei regulären sprachlichen Problemstellungen stehen Wörterbücher zur Verfügung. Da es sich bei der zu behandelnden Textgattung um akademische Texte handelt, sind die zu ergründenden Wörter meistens jedoch nicht in allgemeinen Wörterbüchern erfasst. Folglich muss auf andere Quellen zurückgegriffen werden. Enzyklopädien, Fachbücher und

-texte erklären *per definitionem* den Inhalt eines Konzeptes, jedoch nicht seine sprachliche Verwendung.

5. Es gibt nachgewiesene konkurrierende sprachliche Konstruktionen, die gleichermaßen legitim angewendet werden dürfen

   Bei sprachlichen Wendungen können mehrere konkurrierende Formen gleichzeitig existieren. So können etwa verschiedene Verben mit einem Substantiv kombiniert werden und ungefähr das Gleiche aussagen. Werden unterschiedliche Kombinationen in der entsprechenden (Fach-) Literatur nachgewiesen, müssen sie als legitim erachtet werden. Dementsprechend kann eine Auswahl von mehreren nachgewiesenen sprachlichen Formen nicht als binär, dass heißt als richtig oder falsch, betrachtet werden, sondern muss als kontinuierlich angesehen werden. Eine Entscheidung, welche Form empfohlen und dann auch gegebenenfalls übernommen wird, muss also auf Basis anderer Kriterien getroffen werden. Korpussoftware wie HanConc kann dementsprechend auch nur Hinweise darauf geben, welche sprachliche Form auf Grund von statistischen Merkmalen sprachtypischer für eine bestimmte Disziplin bzw. Textgattung ist (siehe Kapitel 5).

6. Das Imitieren sprachlicher Verwendungen von zu untersuchenden Wörtern ist eine manuelle Form des Text Minings

   Werden, um die kontextuelle Verwendung eines Konzeptes zu ergründen, entsprechende Passagen in Fachtexten konsultiert und die Erkenntnisse auf die zu schreibende Arbeit übertragen, handelt es sich um manuelles Text Mining. HanConc ist ein Versuch, diese manuelle Tätigkeit zu automatisieren und sie somit bei geringerem Aufwand mehr Studierenden und Schreibberater_innen zur Verfügung zu stellen.

7. Sprache kann quantifiziert werden

   Text Mining und quantitative Korpuslinguistik beruhen auf der Annahme, dass mathematisch statistische Aussagen über Sprache getroffen werden können. Zum Beispiel kann eine Einteilungen von unterschiedlichen Texten in Textgattungen auf Basis der Frequenz der beinhalteten Wörter erfolgen (siehe Kapitel 4). Häufigkeiten und Verhältniszahlen unterschiedlicher Wörter können dementsprechend herangezogen werden, um sich zwischen konkurrierenden Konstruktionen zu entscheiden.

8. Korpora sind ein Mittel, kein Zweck

   Korpora sind im Gegensatz zu präskriptiven und dadurch normativen Wörterbüchern deskriptiv. Deswegen bedürfen die Ergebnisse eines verantwortungsvoll programmierten Korpustools einer didaktischen Einbettung und Einordnung.

Kapitel 2

Über den Prozess der Schreibberatung

Dieses Kapitel beleuchtet den Prozess des Anfertigens einer Abschluss- oder Seminararbeit. In einem ersten Schritt geht es um die Teilnehmenden dieses Prozesses und wie sich ihre gegenseitigen Abhängigkeiten unter verschiedenen Gegebenheiten verändern. In einem zweiten Schritt wird der Prozess des Schreibens untersucht. Hier wird beschrieben, wie Korpussoftware den vorher definierten Prozess unterstützen kann. In einem dritten Schritt wird wissenschaftliches Schreiben als Forschungsgegenstand beleuchtet. Diese Unterkapitel stellen in keiner Weise ein neues Schreibmodell auf, sondern formalisieren Beschreibungen aus einschlägiger Schreibberatungsliteratur (Bräuer 2006, Neubauer-Petzoldt 2016, Pydde & Girgensohn 2011, Dengscherz 2020, Spielmann 2011). Auf Grund des Einsatzes als theoretische Fundierung der Schreibberatung der Leibniz Universität Hannover (LUH) wird im Besonderen Grieshammer (2011) und Grieshammer (2013) beachtet. Für eine historische Abhandlung über Schreibberatungsmodelle sei auf Grieshammer (2011) verwiesen.

2.1 Universitäres Schreiben und Schreibberatung als Teil des Ausbildungsprozesses

Dieses Unterkapitel formalisiert die universitäre Ausbildung als Prozess mit mehreren Beteiligten. Unabhängig vom angestrebten Abschluss und den Freiheiten, die etwa eine Studienordnung ermöglicht, sind die abzulaufenden Prozessschritte relativ starr. Lediglich die Bezeichnungen, Quantität und Reihenfolge der Prozessschritte variieren zwischen einem Abschluss in Kunstgeschichte als Bachelorstudiengang oder einem Diplomstudiengang in Quantenphysik.

An dieser Stelle sollen nur die Prozessschritte modelliert werden, die zum Abschluss eines Seminars oder Studiengangs mittels schriftlicher Abschlussarbeit notwendig sind. Als Visualisierung wird eine Swimlane Darstellung gewählt. Diese Darstellung folgt der Idee einer Wettkampfschwimmbahn. Das gesamte Becken symbolisiert die Gesamtorganisation, in diesem Fall eine Universität. Die einzelnen Bahnen repräsentieren Unterorganisationen und Akteure.

Die Swimlanes sind vertikal angeordnet. Die Kopfzeile beinhaltet jeweils den Namen des Akteurs. Ein Akteur kann eine Einzelperson, eine Einzelperson als Repräsentant einer Gruppe oder eine Organisationseinheit sein. Der Prozess startet jeweils im Rechteck in der oberen linken

Swimlane und verläuft dann über mehrere Ereignisse (Rechtecke) und Entscheidungen (Rauten; mit Pfaden nach unten für *Ja* und Pfaden zur Seite für *Nein*) zu einem abschließenden Rechteck. Hierbei folgt die Leserichtung den Pfeilen nach unten. Einzelne Schritte können rekursiv und iterativ, d.h. mehrfach abgeschritten werden. Alle Schritte sind nummeriert. Folgen auf eine Entscheidung unterschiedliche Prozesschritte, werden diese bis zum erneuten Zusammenführen mit Buchstaben nummeriert.

2.1.1 Schreiben ohne Unterstützung

Die erste Darstellung (Abbildung 2.1) visualisiert den Schreibprozess für den Fall, dass die Studierende ohne weitere Hilfe die Arbeit verfasst und die fachwissenschaftlich Betreuende auch gleichzeitig die Arbeit bewertet. Als einzige mögliche Unterstützung dienen Freunde, Verwandte und Mitstudierende, welche eine Rückmeldung zur Arbeit liefern könnten.

Der Startpunkt (1) ist durch die Notwendigkeit eine Arbeit zu verfassen definiert. Bei einer Arbeit kann es sich um jedwede schriftliche Leistung, die im Rahmen eines Studiums anfallen kann, handeln. Es kann sich also um Projekt- und Abschlussarbeiten, genauso gut aber auch um Laborberichte, Protokolle von Experimenten oder Ähnliches handeln.

Schritt (2) ergibt sich aus der Art der Arbeit. Bei Laborberichten und Protokollen ergibt sich das Thema aus dem vorherigen Experiment oder dem Kursthema. Andernfalls kann das Thema durch die betreuende Professor_in veröffentlicht oder in einem persönlichen Gespräch festgelegt werden.

Die Strukturbesprechung (3) richtet sich nach dem Typus der Arbeit, den Vorschriften der betreuenden Organisationseinheit und/oder dem Gespräch mit der Betreuenden. Die Fakultät für Maschinenbau der LUH befindet zum Beispiel, dass „[es] neben der üblichen Betreuung [...] ratsam [ist], nach dem ersten Drittel der Bearbeitungszeit ein ausführliches Feedbackgespräch zu führen." Verpflichtend ist dieses Gespräch allerdings nicht. Für den Fall, dass dieses Gespräch nicht stattfindet und die Gliederung nicht explizit durch das betreuende Institut oder den Betreuenden in Form eines Dokumentes oder einer Word/LaTex Vorlage vorgegeben wird, liegt es alleine am Schreibenden, eine sinnvolle Gliederung des Gesamttextes und seiner Untereinheiten zu finden. Da ingenieurwissenschaftliche Studienordnungen im Vergleich zu geisteswissenschaftlichen Studienordnungen nur wenige gleichartige schriftliche Leistungen erwarten, ist auch nicht davon auszugehen, dass Studierende ingenieurwissenschaftlicher Studienfächer bereits eine gleichartige Arbeit verfasst, abgegeben und bewertet bekommen haben (siehe Kapitel 3). Somit haben sie keine Möglichkeit auf den Erfahrungen aus einer vorherigen Arbeit aufzubauen. Es ist davon auszugehen, dass sich viele Studierende dieser Fachrichtungen daher mit einer höheren Wahrscheinlichkeit nicht an wissenschaftliche Standards halten, als solche Studierende, deren Studium zu einem größeren Teil aus gleichartigen schriftlichen Leistungen besteht.

Die Literaturrecherche (4) ergibt sich aus den Schritten (1) bis (3). Je nach Art der Arbeit ist sie unterschiedlich intensiv. Arbeiten etwa, die sich auf eigene Experimente oder Projekte beziehen, brauchen weniger Literaturverweise als theoretisch gehaltene Arbeiten.

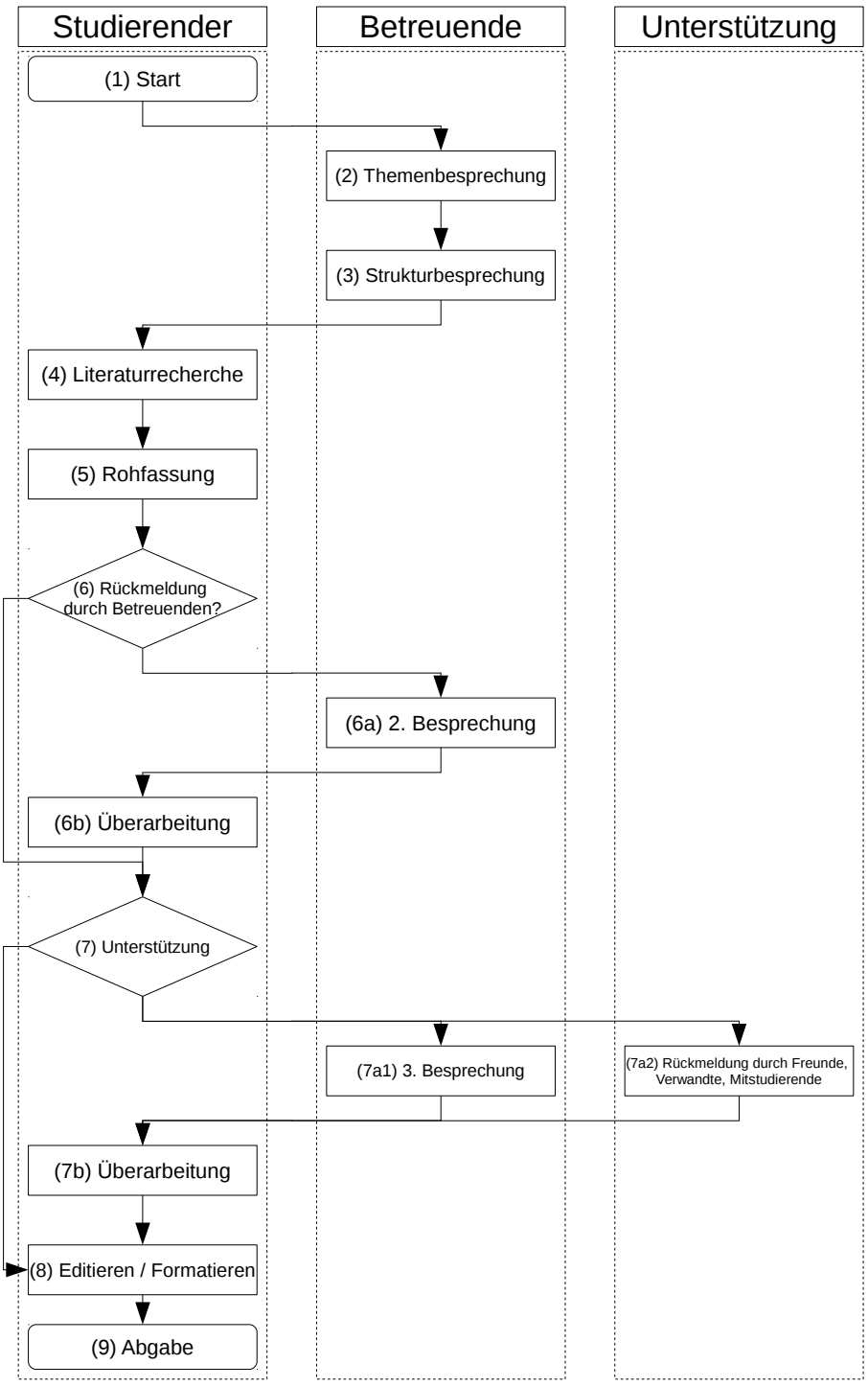


Abbildung 2.1: Swimlane Darstellung eines einfachen Schreibprozesses

Je nach Art der Arbeit wird Schritt (5) einzeln oder als Gruppe durchgeführt. Da in dieser Arbeit vornehmlich von Abschlussarbeiten ausgegangen wird und diese selbstständig verfasst werden müssen (vgl. etwa §7 (5) a der Prüfungsordnung für den Bachelorstudiengang Produktion und Logistik vom 2. August 2017), wird Schritt (5) als Einzelarbeit angenommen.

Schritt (6) und (7a1) sind abhängig vom Betreuungsgrad zwischen Schreibender und Betreuender. Findet außer dem ersten Gespräch (2) und der Abgabe (9) keine weitere Betreuung statt, so werden die Schritte (6) und (7a1) übersprungen. Andernfalls können die Schritte (6), (6a), (6b), (7) und (7a1) durchaus mehrfach durchlaufen werden. Die Hinweise zur Bewertung von Studien- und Abschlussarbeiten etwa der Fakultät für Maschinenbau der LUH sehen die Schritte unter (6) und (7) explizit vor. Da diese Schritte jedoch nicht verpflichtend sind, kann es dazu kommen, dass die Betreuende die Arbeit das erste Mal nach der Abgabe lesen kann. Schritt (6) beinhaltet nur eine Rückmeldung zum fachwissenschaftlichen Inhalt der Arbeit.

Schritt (7) bezieht sich im Gegensatz zu Schritt (6) rein auf die äußere Gestalt der Arbeit. Hierunter fallen Rechtschreibung, Grammatik, logischer Aufbau innerhalb eines Kapitels und Formalia wie das Einhalten von Zitierweisen. Dieser Schritt bedarf nicht unbedingt einer inhaltlichen Auseinandersetzung mit der Arbeit. Es ist zu betonen, dass es für ausländische Studierende, deren Muttersprache nicht Deutsch ist und denen ein Netzwerk aus Unterstützer_innen mit ausreichenden Deutschkenntnissen fehlt, ungleich schwerer ist, entsprechend hilfreiche Rückmeldungen zu erhalten.

Schritt (8) umfasst das Erstellen der Inhalts-, Literatur- und Abbildungsverzeichnisse, das Anpassen der Word- bzw. LaTex-Dokumente und die Gestaltung des Deckblattes.

In einem letzten Schritt erfolgt die Abgabe der Arbeit (9). Je nach Studiengang ist während des gesamten Schreibprozesses eine einzige Besprechung der Studierenden mit ihrer Betreuenden ausreichend. In diesem Fall bestünde die einzige Rück-meldung aus den Kommentaren an der abgegebenen Arbeit und der zu erwartende Lerneffekt wäre minimal. Es ist an dieser Stelle unbekannt, ob die Studierende die Arbeit nach der Bewertung des Betreuenden zurück erhält und wie sehr die Betreuende auf sprachliche, stilistische und grammatikalische Rückmeldungen Wert legt.

Kapitel 3 wird zeigen, dass Studierende nicht-geisteswissenschaftlicher Studiengänge kaum Möglichkeiten haben, schriftliche Leistungen zu erbringen. Die Abschlussarbeiten hingegen haben in diesen Studiengängen ein ungleich höheres Gewicht im Verhältnis zu allen schriftlich zu erbringenden Leistungen.

2.1.2 Schreiben mit institutionalisierter Hilfe

Im Gegensatz zum Schreibprozess in Abbildung 2.1 beinhaltet der im Folgenden beschriebene Schreibprozess institutionalisierte Unterstützung (rechte Swimlane). Hierbei kann es sich etwa um universitäre Einrichtungen, Studierendengruppen oder private Anbieter für Unterstützungsangebote handeln.

Das Schaubild ist an mehreren Stellen iterativ angelegt, sodass diese Schritte mehrfach durchlaufen werden können. Iterative Schritte sind als Entscheidung dargestellt und mit einer

Raute (#) nummeriert. Die Entscheidung, ob genügend Iterationen durchlaufen wurden, obliegt der Studierenden in Absprache mit der Schreibberater_in. Kapitel 2.1.4 beschreibt diesen Teil genauer.

Die Schritte (1) bis (3) und (4) bis (9) sind analog zu Abbildung 2.1 angelegt. Der folgende Schritt führt allerdings nicht direkt zur Literaturrecherche (4), sondern zur Vorbesprechung, bzw. der ersten Strukturbesprechung. Es soll an dieser Stelle von zwei idealisierten Prämissen ausgegangen werden:

- Studierende, deren Muttersprache nicht die Zielsprache der zu verfassenden Arbeiten ist, lernen im Verlauf ihres Studiums einzuschätzen, wie gut ihre Fähigkeiten in Bezug auf die Zielsprache und wissenschaftliches Schreiben sind.
- Zudem wird angenommen, dass Studiengänge in Bezug auf die wissenschaftliche Herausforderung ansteigend ausgelegt sind und die hierfür notwendigen Anforderungen schriftlich überprüft werden.

Studierende, die institutionalisierte Hilfe brauchen oder wünschen, bemerken dies im Idealfall bereits frühzeitig, sodass die Hilfe parallel zu Beginn der Arbeitsphase anlaufen kann.

In der unterstützenden Vor- und Strukturbesprechung mit einer Schreibberater_in haben Studierende einen doppelten fachlichen Informationsvorsprung. Auf Basis ihres Studiums haben sie die notwendigen Vorkenntnisse, um den Gegenstand der Arbeit einzuordnen und zu durchdringen. Durch ihre eigene Vorbesprechung mit der fachlich Betreuenden (3) und ggf. durch die Ausschreibung der Arbeit, sind sie mit den fachlichen, technischen und inhaltlichen Anforderungen bereits grundlegend vertraut. Die erste Vorbesprechung mit der Schreibberater_in sollte daher genutzt werden, um diesen Vorsprung auszugleichen, indem die Studierenden ihren Schreibberater_innen einen fachlichen Überblick zum Thema der Arbeit geben.

Grieshammer (2013) schlagen in „Zukunftsmodell Schreibberatung" vor, dass sich Schreibberater_innen statt mit dem Inhalt der betreuten Arbeit nur mit der Unterstützung eines allgemeineren Schreibprozesses beschäftigen. Kapitel 2.1.4, 2.1.5 und 3 beschreiben, warum eine detailliertere Beschäftigung mit den Arbeiten der Studierenden notwendig ist und somit der Vorschlag von Grieshammer (2013) nicht praktikabel erscheint.

2.1.3 Unterstützende Vor- und Strukturbesprechung

Strukturbesprechungen sind das Herzstück der Schreibberatung. Unter Struktur soll hier der Inhalt einer fachlichen Arbeit im Sinne von Begründungsmustern, Textaufbau, Syntax, äußerer Form und Rechtschreibung, Grammatik und Semantik gefasst werden. Je nach Beratungsschwerpunkt sind die oben genannten Facetten zu gewichten. Jede dieser Facetten kann, wenn sie schwach ausgeprägt ist, das Verständnis eines Textes massiv behindern, wenn nicht sogar verhindern. Auch wenn diese Erkenntnis trivial erscheint, so hat sie doch zur Folge, dass eine Schreibberatung für Studierende, die in einer Fremdsprache schreiben, nicht erfolgreich sein kann, wenn sie sich vom Inhalt und den linguistischen Bestandteilen zurückzieht.

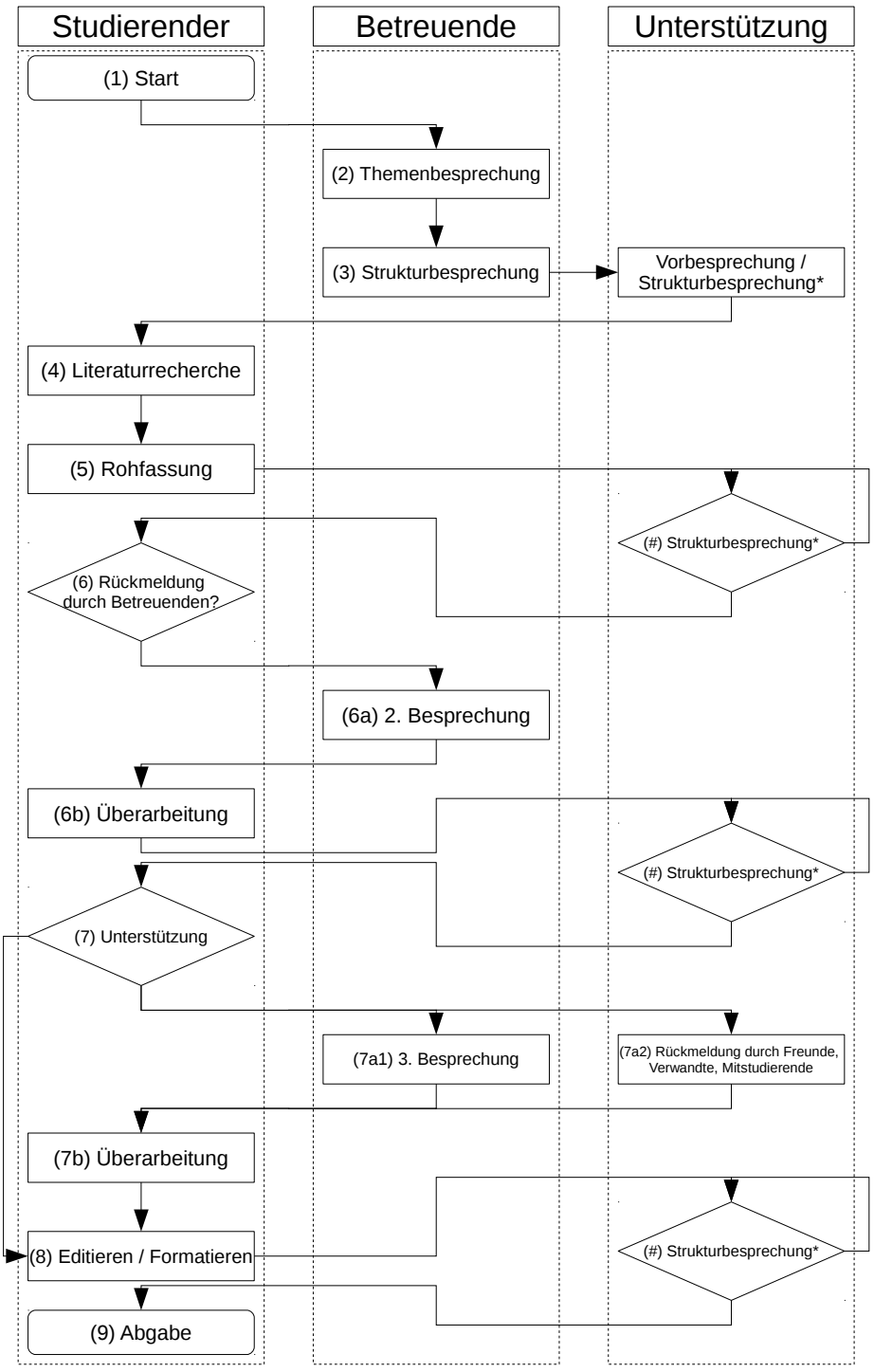


Abbildung 2.2: Swimlane Darstellung eines Schreibprozesses mit institutionalisierter Hilfe

Die obige Argumentation soll an einem Beispiel verdeutlicht werden. Thema der Arbeit sei ein Biegeversuch aus der Werkstoffkunde nach DIN EN 7438-2012. Hierbei wird ein Stück des Werkstoffs[1] auf zwei Auflagen gelegt und mittig mit einem Prüfstempel belastet. Das Werkstück wird nun bis zum Bruch mit zunehmender Kraft belastet. Hierdurch können die Materialeigenschaften des Werkstoffs ermittelt werden. Der Satz: „Bei dem Drei-Punkt Biegeversuch wurde eine Bruchkraft von 3.025 N ermittelt." wird durch Abschwächung der entsprechenden Facetten in den unten stehenden Sätzen verändert wiedergegeben:

- **Begründungsmuster:**
  „Eine Bruchkraft von 3.025 N wurde in dem Drei-Punkt Biegeversuch festgesetzt."
- **Textaufbau:**
  „Der Werkstoff hat eine Bruchkraft von 3.025 N. Es wurde ein Drei-Punkt Biegeversuch durchgeführt."
- **Syntax:**
  „Eine Bruchkraft von Drei-Punkt Biegeversuch ermittelt 3.025 N."
- **Äußere Form:** Ein auf Grund der Schriftart, der Schriftgröße, des Schriftsatzes oder der Druckqualität unlesbares Dokument.
- **Rechtschreibung und Grammatik:** „Bei dem ... Biegefersug wurde ein Buchkraft von 3.025 N gemittelt."[2]
- **Semantik:**
  „Bei dem Drei-Punkt Biegeversuch wurde eine Bruchkraft von 3.025 N geschätzt."

Trotz der Abweichungen sind alle Sätze mit Hilfe der oben stehenden Einleitung verständlich. Erst durch eine Kombination mehrerer Facetten und ohne ausreichend Kontext wird der Satz vollends unverständlich. Wird die gesamte Arbeit unstrukturiert mit solchen Sätzen verfasst, so ist davon auszugehen, dass sie nicht von der Betreuenden akzeptiert oder zumindest nicht mit einer für die Studierende zufriedenstellenden Note bewertet wird. Es ist daher notwendig alle diese Facetten in einer Strukturbesprechung zu berücksichtigen.

2.1.4 Entscheidungsgrundlagen für die Iterationsanzahl der Strukturbesprechungen

Eine Schreibberatung kann sich auf eine oder mehrere der im vorherigen Kapitel erläuterten Facetten beziehen (siehe Kapitel 2.1.3). Zur Ermittlung der Anzahl der notwendigen Schreibberatungen soll ein mathematisches Modell formuliert werden. Die Fähigkeit der Studierenden

[1] In diesem Fall ein Stück Metall.

[2] Das obige Beispiel mit den drei Punkten wirkt absurd. Wird jedoch stattdessen das Beispiel einer Instrumentalvariablen (IV) herangezogen und die Abkürzung als römische Zahl missverstanden, so kann schnell aus einer IV Regression eine 4 Regression werden, die nur noch mit umfangreichem Kontextwissen als solche erkannt werden kann.

und die Qualität ihrer Arbeit in Bezug auf die oben genannten Facetten soll auf einer 6 stufigen Likert Skala[3] geschätzt werden (Bortz 2010). Wie die Parametrisierung und die darauf basierende Einstufung erfolgen sollen, ist für diese Arbeit irrelevant.

Die Schreibberatung hat zum Ziel, den Likert-Wert mindestens einer dieser Facetten zu erhöhen. In Bezug auf die Studierenden erfolgt dies direkt durch Anleitung durch die Schreibberater_in. In Bezug auf die Arbeit wird der Wert durch die verbesserten Fähigkeiten der Studierenden und die erfolgten Anpassungen am Text erreicht. Liegt etwa der Wert für die Facette „Grammatik" vor einer Beratung bei 4, kann dieser durch eine Anleitung zum gezielten Grammatiktraining innerhalb weniger Sitzungen auf 5 gehoben werden. Verallgemeinert lassen sich die oben genannten sechs Facetten als Vektor darstellen:

$$F\ddot{a}higkeiten_{Studierende} = \begin{pmatrix} F_0 & F_1 & F_2 & F_3 & F_4 & F_5 \end{pmatrix} \tag{2.1}$$

mit F_i = Fähigkeit i, wobei jedes Element eine Facette darstellt. Ausgehend davon, dass die Studierende der einzige Einflussfaktor auf die Qualität der Arbeit und damit auf die Benotung ist und die übrigen Einflussfaktoren (siehe etwa Abbildung 2.2) nur über die Studierenden einfließen, würde gelten:

$$F\ddot{a}higkeiten_{Studierende} = Qualit\ddot{a}t_{Arbeit} \tag{2.2}$$

.

Da jedoch der Inhalt einer Arbeit für die Benotung entscheidend ist und etwa eine besonders gute Grammatik höchstens implizit in die Benotung einfließt, muss angenommen werden, dass erst eine mangelhafte Grammatik explizit wahrgenommen wird und so zu einer Abwertung führt. Daher gilt:

$$\min F\ddot{a}higkeiten_{Studierende} = Qualit\ddot{a}t_{Arbeit} \tag{2.3}$$

.

Aus Formel 2.3 ergibt sich die Vektordarstellung 2.4. Es wird davon ausgegangen, dass sich die einzelnen Elemente nicht gegenseitig aufwiegen lassen. Das bedeutet, dass beispielsweise eine besonders hoch ausgeprägte Grammatikfähigkeit eine besonders schwach ausgeprägte Rechtschreibfähigkeit nicht ausgleichen kann.

Die fachlichen und didaktischen Fähigkeiten der Schreibberater_in (ϕ) führen innerhalb einer Schreibberatung zu einer Veränderung des obigen Fähigkeiten-Vektors, sodass:

$$F\ddot{a}higkeiten^{DurchBeratung}_{Studierende} = \begin{pmatrix} F_0 \cdot \phi_0 & F_1 \cdot \phi_1 & F_2 \cdot \phi_2 & F_3 \cdot \phi_3 & F_4 \cdot \phi_4 & F_5 \cdot \phi_5 \end{pmatrix} \tag{2.4}$$

mit F_i = Fähigkeit i; ϕ_i = Fähigkeiten der Schreibberater_in

[3]Ein Wert von 1 entspricht hier der geringsten Ausprägung einer Facette und ein Wert von 6 der höchstmöglichen Ausprägung.

Weil allerdings nicht jede Facette in jeder Schreibberatung berücksichtigt werden kann, wird Gleichung 2.4 erweitert zu:

$$Fähigkeiten_{Studierende}^{DurchBeratung} = \begin{pmatrix} (F_0 \cdot \phi_0 \cdot x_0) + F_0 \\ (F_1 \cdot \phi_1 \cdot x_1) + F_1 \\ (F_2 \cdot \phi_2 \cdot x_2) + F_2 \\ (F_3 \cdot \phi_3 \cdot x_3) + F_3 \\ (F_4 \cdot \phi_4 \cdot x_4) + F_4 \\ (F_5 \cdot \phi_5 \cdot x_5) + F_5 \end{pmatrix} \tag{2.5}$$

mit $F_i =$ Fähigkeit i; $\phi_i =$ Fähigkeit i der Schreibberater_in; $x_i =$ boolescher Schalter i für Inhalt einer Schreibberatung.

Die x Variable ist in diesem Fall boolesch mit $x = 1$, falls die Facette behandelt wurde und anderenfalls $x = 0$. Dies führt dazu, dass sich nur die bearbeiteten Fähigkeiten verändern.

Jede Schreibberatung verändert die Fähigkeiten der Studierenden und mittelbar die Qualität der Arbeit. Die einzelnen Elemente des Fähigkeiten-Vektors ändern sich entsprechend der Vorbelegung, des Beratungsfokus' der Sitzung und der Fähigkeiten der Schreibberater_in. Um nun zur ermitteln, ob ausreichend viele Schreibberatungen in Anspruch genommen wurden, muss für jedes Element des Zielvektors ein angenommener Minimalwert bestimmt werden. Dieser Zielwert entspricht, je nach Selbstanspruch der Studierenden, den gestaffelten Anforderungen der fachlich Betreuenden. Die Anforderungen können nur geschätzt werden, da selbst bei kleinteiligsten Bewertungsvorgaben persönliche Einschätzungen zu intersubjektiven Unterschieden führen (Trapmann, Hell, Weigand & Schuler 2007). Bezogen auf die vorherige Notation ergibt sich also:

$$Fähigkeiten_{Studierende}^{DurchBeratung} = \begin{pmatrix} \epsilon_0 = Q_0 - ((F_0 \cdot \phi_0 \cdot x_0) + F_0) \\ \epsilon_1 = Q_1 - ((F_1 \cdot \phi_1 \cdot x_1) + F_1) \\ \epsilon_2 = Q_2 - ((F_2 \cdot \phi_2 \cdot x_2) + F_2) \\ \epsilon_3 = Q_3 - ((F_3 \cdot \phi_3 \cdot x_3) + F_3) \\ \epsilon_4 = Q_4 - ((F_4 \cdot \phi_4 \cdot x_4) + F_4) \\ \epsilon_5 = Q_5 - ((F_5 \cdot \phi_5 \cdot x_5) + F_5) \end{pmatrix} \tag{2.6}$$

mit $F_i =$ Fähigkeit i; $\phi_i =$ Fähigkeit i der Schreibberater_in; $x_i =$ boolescher Schalter i für Inhalt einer Schreibberatung; $Q_i =$ erwartete Qualitätsanforderung des fachlich Betreuenden; $\epsilon_i =$ Differenz aus den Anforderungen und den bereits erbrachten Leistungen
Hieraus folgt, dass:

$$Beratungsbedarf(\epsilon) = \min Fähigkeiten_{Studierende}^{DurchBeratung}; \quad \epsilon \; in - \infty \to 0 \tag{2.7}$$

Es folgt also, dass Schreibberatungen solange in Anspruch genommen werden müssen, bis die Differenz zwischen der Qualität der Arbeit und den erwarteten Anforderungen durch die fachlich Betreuende mindestens null beträgt oder positiv ist[4]. Erst ab diesem Zeitpunkt kann von

[4]Es ist anzumerken, dass, wie oben beschrieben, weder die Fähigkeiten der Studierenden noch der Schreibbera-

einer bestandenen Arbeit ausgegangen werden. Da es sich hierbei um Minimalanforderungen handelt, können durchaus weitere Termine wahrgenommen werden, um die Wahrscheinlichkeit für eine Leistungsverbesserung und damit für eine bessere Note zu erhöhen. Die Frage, ob es zu viele Schreibberatungen geben kann, soll hier nicht erörtert werden. Es ist aber davon auszugehen, dass Abgabetermine und kapazitative Beschränkungen der Schreibzentren natürliche Grenzen darstellen.

Darauf aufbauend erläutert das folgende Kapitel, wie Korpuslinguistik in dieses Modell integriert werden kann. Kapitel 5 wird diese Einsatzmöglichkeiten präzisieren und zur praktikablen Anwendung aufbereiten.

2.1.5 Korpuslinguistische Unterstützung in der Schreibberatung

Dieses Kapitel erweitert das oben stehende mathematische Modell um korpuslinguistische Software. Die Effektivität dieser Software soll in mehreren Dimensionen dargestellt werden. Diese Dimensionen gehen über die reine Didaktik hinaus und beinhalten etwa Rahmenbedingungen eines Schreibzentrums. Auf einzelne Aspekte dieser Dimensionen wird in späteren Kapiteln detaillierter eingegangen.

Als Endanwender_innen für korpuslinguistische Software in der Schreibberatung kommen sowohl Schreibberater_innen als auch Studierende in Frage[5]. Die Wahl der Endanwender_innengruppe hat nicht nur Auswirkungen auf die schreibdidaktische Herangehensweise sondern auch konkrete Folgen für das Design der Korpussoftware.

Es wird nachfolgend auf fünf Dimensionen eingegangen, denen Hannover Concordancer (HanConc) angepasst werden muss. Es soll auch ein Vorschlag für eine Zuordnung der Dimensionen zu den Endanwender_innen vorgestellt werden. Allerdings können lokale Gesetze, Vorschriften der hostenden Einrichtung, technische Möglichkeiten, d.h. Serverkapazitäten, linguistische Kenntnisse der Endanwender_innen und didaktische Entscheidungen zu deutlichen Abweichungen zu den hier dargestellten Dimensionen führen.

Da jedes Forschungsvorhaben als individuell betrachtet wird, muss HanConc auf die jeweiligen Bedürfnisse der Forschenden speziell abgestimmt und ggf. umprogrammiert werden. Werden eigene Daten und Hardware verwendet, so obliegt die Einhaltung von Vorschriften und Gesetzen sowie die Erstellung eines Sicherheitskonzeptes, soweit notwendig, den entsprechenden Forscher_innen.

Die folgenden Dimensionen müssen von Schreibzentren bei der Einrichtung von Korpussoftware und HanConc im Speziellen bedacht werden:

---

ter_in gemessen werden. Ebenso werden die Erwartungen der fachlich Betreuenden nicht empirisch erhoben. Die Erwartungen können daher nur in persönlichen Gesprächen von der Studierenden mit der fachlich Betreuenden approximiert werden.

[5]Eine dritte Gruppe könnten Korpuslinguist_innen darstellen. Diese werden als Endanwender_innengruppe in diesem Kapitel nicht im Fokus stehen. Es wird davon ausgegangen, dass sie sich mit dem Quellcode auseinandersetzen oder eines der Frontends benutzen. Der Quellcode wurde möglichst modular gestaltet, sodass es von vornherein vorgesehen ist, dass er verändert und für eventuelle Forschungsprojekte angepasst werden kann.

Dimension I: Linguistische und technische Fähigkeiten der Schreibberater_innen
Schreibberater_innen wird grundsätzlich zugeschrieben, dass sie über ein linguistisches Grundvokabular verfügen, aber nicht zwangsläufig programmieren können. Es ist also nicht notwendig, komplett auf linguistisches Fachvokabular zu verzichten. Das Frontend kann jedoch so designt werden, dass entsprechend komplexere Suchanfragen formuliert werden können.

Dimension II: IT Architektur
Soll HanConc Studierenden oder Schreibberater_innen zur Verfügung gestellt werden, so kann dies auf drei Arten erfolgen:

Sofern die Schreibberatung institutionell organisiert ist, ist es möglich, eine zentralisierte IT Architektur zu wählen, sodass es keiner Datenhaltung auf den Endgeräten der Studierenden oder Schreibberater_innen bedarf. Somit ist es auch möglich, die Administration zu zentralisieren und damit eine höhere Verfügbarkeit zu erreichen. Die Software kann zentral gehostet und per Webzugang[6] zur Verfügung gestellt werden. Alternativ kann HanConc auch direkt auf der Hardware der Studierenden[7] oder Schreibberater_innen installiert werden.

Da HanConc, um schnelle Abfragezeiten zu ermöglichen, alle Korpusdaten im RAM hält, kann es bei einem großen Korpus in Kombination mit limitierter Endkunden-Hardware zu Systemausfällen und -abstürzen kommen. Dies ist vor allem bei der Installation auf den Computern von Studierenden und Schreibberater_innen zu beachten. Als Beispiel wird ein Doppelkernprozessor mit 4 GB RAM und einem 1 GB Korpus angenommen. Die Programmiersprache R, in der HanConc zum Großteil geschrieben ist, verdoppelt etwa die Größe der eingelesenen Dateien im RAM im Vergleich zur Festplatte. Sind gleichzeitig weitere Programme geöffnet, kann es dazu führen, dass der RAM vollläuft und einzelne Programme abstürzen. Führt dies zu Datenverlust, kann das die Akzeptanz von HanConc als Hilfsmittel bei den Schreibberater_innen mindern.

Wird HanConc zentral gehostet, können die Hardwarerestriktionen einfacher überwunden werden. Die HanConc zugrunde liegende Technologie ist vertikal, durch einen stärkeren Server, und horizontal, durch eine höhere Anzahl an Servern, skalierbar. Durch den Einsatz von vorgeschalteten Load Balancern[8] und Containerisierung[9] kann HanConc beliebig erweitert werden. Kapitel 5 geht auf die Hardwareanforderungen und Architekturvorschläge genauer ein.

Dimension III: Passgenauigkeit des Frontends
Unabhängig davon wie HanCon Studierenden oder Schreibberater_innen zur Verfügung gestellt wird, muss das Frontend in Bezug auf linguistisches Fachvokabular und Suchkomplexität angepasst werden. Bei den meisten Studierenden, die an der LUH eine Schreibberatung in Anspruch nehmen können, muss angenommen werden, dass sie in ihrem Studium keine

[6] Dieser Zugang kann auf einen Computerraum limitiert sein oder mit entsprechenden Authentifizierungsmechanismen über das Internet zugänglich sein.

[7] In diesem Fall ergeben sich Fragen bezüglich des Urheberrechts, da beispielsweise nicht kontrolliert werden kann, ob die Dokumente weitergegeben oder nach Beendigung der Schreibberatungen gelöscht werden.

[8] Bei Load Balancern handelt es sich um Server, die den eigentlich Anwendungsservern vorgeschaltet sind, um die Anfragen gleichmäßig zu verteilen und so eine Überlastung einzelner Server zu verhindern.

[9] Wird HanConc etwa in einem Docker Container auf einem Server gestartet, so kann durch die weitere Abstraktionsschicht die vorhandene Hardware besser ausgelastet und flexibler auf Last durch das Spawnen von zusätzlichen Containerinstanzen reagiert werden.

Berührungspunkte zur Linguistik haben (vgl. Kapitel 3). Entsprechend muss das Frontend vereinfacht werden und ein größerer Teil der Suchlogik statisch in das Backend übertragen werden.

Dimension IV & V: Compliance und Sicherheitsanforderungen
Eher administrativen Charakter haben die übrigen zwei Dimensionen. Nationale Gesetzte als auch Vorschriften der jeweiligen Universitäten regeln den Umgang mit geistigem Eigentum an der Software als auch der zugrunde liegenden Texte. Um die Einhaltung dieser Regelungen zu gewährleisten, sollte ein entsprechendes Sicherheitskonzept etwa nach ISO 2700x oder BSI IT-Grundschutzkompendium entwickelt und angewendet werden (Bundesamt für Sicherheit in der Informationstechnik 2020, Disterer 2013). Nur so kann ein Abgreifen der Texte verhindert werden.

Die oben genannten Dimensionen beeinflussen direkt die Flexibilität und Effektivität von Korpussoftware. Linguistisch geschulte Programmierer_innen können aus einer lokalen Installation von HanConc sicherlich einen höheren Nutzen ziehen, als linguistische Laien mit einer online Version, die nicht auf ihre Bedürfnisse abgestimmt ist. Zusammengefasst lässt sich die Effektivität der Dimensionen wie folgt abbilden:

$$\begin{aligned}\text{Effektivität}_{\text{HanConc}} = \;& \text{Effektivität der Endanwender\_in} \cdot \\ & \min \text{linguistische Fähigkeit der Teilnehmer\_innen der Schreibberatung} \cdot \\ & \text{Passgenauigkeit des Frontends} \cdot \\ & \text{Compliance} \cdot \\ & \text{Sicherheitsanforderungen} \cdot \\ & \text{IT Architektur}\end{aligned} \tag{2.8}$$

Es ist darauf hinzuweisen, dass eine Nichtbeachtung schon einer dieser Dimensionen zu einem kompletten Effektivitätsverlust von HanConc führen kann. An drei Beispielen soll dies verdeutlicht werden:

- Software, die gegen geltende Gesetzte verstößt, darf nicht eingesetzt werden. Sollten Gesetze Text Mining von urheberrechtlich geschützten Texten verbieten, kann auch HanConc nicht wie vorgesehen in der Schreibberatung eingesetzt werden.
- Ist das Frontend auf wissenschaftlich forschende Korpuslinguist_innen abgestimmt, kann es für eine Studierende ohne linguistische Kenntnisse und mit geringen Fähigkeiten der Sprache des Frontends kaum oder gar nicht hilfreich sein.
- Bei einer auf einem Server gehosteten Version von HanConc kann eine mangelhafte Skalierung zur Unbenutzbarkeit führen.

Formel 2.6 soll nun um HanConc erweitert werden. Hierzu wird der Term aus Formel 2.8 auf der rechten Seite eingesetzt. Formel 2.7 bleibt davon unberührt.

$$Fähigkeiten_{Studierende}^{DurchBeratung} = \begin{pmatrix} \epsilon_0 = Q_0 - ((F_0 \cdot \phi_0 \cdot x_0 + Effektivität_{HanConc/0}) + F_0) \\ \epsilon_1 = Q_1 - ((F_1 \cdot \phi_1 \cdot x_1 + Effektivität_{HanConc/1}) + F_1) \\ \epsilon_2 = Q_2 - ((F_2 \cdot \phi_2 \cdot x_2 + Effektivität_{HanConc/2}) + F_2) \\ \epsilon_3 = Q_3 - ((F_3 \cdot \phi_3 \cdot x_3 + Effektivität_{HanConc/3}) + F_3) \\ \epsilon_4 = Q_4 - ((F_4 \cdot \phi_4 \cdot x_4 + Effektivität_{HanConc/4}) + F_4) \\ \epsilon_5 = Q_5 - ((F_5 \cdot \phi_5 \cdot x_5 + Effektivität_{HanConc/5}) + F_5) \end{pmatrix} \tag{2.9}$$

mit $F_i =$ Fähigkeit i; $\phi_i =$ Fähigkeit i der Schreibberater_in; $x_i =$ boolescher Schalter i für Inhalt einer Schreibberatung; $Q_i =$ erwartete Qualitätsanforderung der fachlich Betreuenden; $\epsilon_i =$ Differenz aus den Anforderungen und den bereits erbrachten Leistungen;
$Effektivität_{HanConc/i} =$ Effektivität von HanConc in Bezug auf Fähigkeit i.

HanCon hat, richtig eingesetzt, somit einen reduzierenden Effekt auf die Iterationszahl an Beratungssitzungen.

2.1.6 Vorschläge zur empirischen Überprüfung des korpuslinguistischen Schreibberatungsmodells

Das oben beschriebene Modell erläutert, wie Schreibberatung und Korpussoftware Studierende beim Verfassen ihrer Arbeiten unterstützen kann. Die Annahmen, die diesem Modell zugrunde liegen, können empirisch überprüft werden. Es soll an dieser Stelle jedoch nur das Vorgehen für eine solche Überprüfung skizziert werden; eine tatsächliche Überprüfung würde durch ihren Umfang den Rahmen dieser Arbeit überschreiten und kann in einem anderen Forschungsprojekt ausgeführt werden.

Zunächst müsste gezeigt werden, ob und wie die linguistische Qualität einer studentischen Arbeit Einfluss auf deren Erfolgswahrscheinlichkeit hat. Hierzu muss als Datengrundlage zuerst ein Korpus erstellt werden. Dieses Korpus muss Texte unterschiedlicher Textsorten, d.h. etwa Praktikumsberichte, Bachelor- und Masterarbeiten, aus verschiedenen Fachrichtungen enthalten. Zusätzlich bedarf es einer Tabelle, die Informationen über die Benotung dieser Arbeiten enthält. Hieraus ergibt sich ein Klassifikationsproblem mit den beiden Klassen „hat bestanden" und „hat nicht bestanden". Ein entsprechendes Modell für dieses Klassifikationsproblem würde, basierend auf den linguistischen Features der Texte, die Klasse vorhersagen. Das obige Modell kann nun überprüft werden, indem vor und nach jeder Schreibberatung der Text klassifiziert wird. Mittels einer Vergleichsgruppe kann getestet werden, ob Studierende, die HanConc nutzen, einen wahrscheinlich „bestandenen" Text schneller produzieren können. Als weitere Vergleichsgruppen müssten Studierende ohne Schreibberatung und Studierende mit Schreibberatung herangezogen werden.

Um die beschriebenen linguistischen Features zu ermitteln, könnten Forschungsergebnisse zu textueller Komplexität verwendet werden. Ansätze diese Komplexität hochskaliert automatisch zu analysieren, gibt es seit einigen Jahren. Hier sollen zwei Ansätze vorgestellt werden:

Coh-Metrix analysiert Texte in Bezug auf mehr als einhundert Parameter. Es ist zu beachten, dass einige Parameter hoch miteinander korrelieren und daher die absolute Zahl an nutzbaren Parametern deutlich geringer ist. Die Parameter lassen sich in drei Kategorien einteilen (McNamara, Graesser, McCarthy & Cai 2014):

1. **Parameter beruhend auf Häufigkeiten in der Ausgangssprache**
   Die Häufigkeit der Wörter im allgemeinen Sprachgebrauch und ihre Exklusivität in Bezug auf einen bestimmten Kontext sorgen dafür, dass bestimmte Wörter eher als „akademisch" oder„allgemeinsprachlich" wahrgenommen werden. Das Wort „ersuchen" zum Beispiel erzeugt andere Konnotationen als „bitten", obwohl damit der gleiche Vorgang gemeint ist. Allerdings ergibt sich für das Deutsche im Allgemeinen und das technisch-akademische Schreiben im Speziellen das Problem, dass entsprechend anerkannte Korpora zum jetzigen Zeitpunkt nicht zur Verfügung stehen.

2. **Parameter beruhend auf Algorithmen zur Textaufbereitung**
   Syntaktische Komplexität kann nur unter Zuhilfenahme eines entsprechenden Parsers oder Taggers verwendet werden. Der in McNamara et al. (2014) verwendete Parser sowie Helmut Schmids TreeTagger (Schmid 1994, Schmid 1999) und der Stanford NLP Parser (Manning & Schutze 1999) basieren noch auf Shallow Learning Algorithmen wie Decision Trees und Markov Modellen. Neuere Modelle, die auf neuronalen Netzwerken basieren und eine deutlich höhere Genauigkeit versprechen (Andor, Alberti, Weiss, Severyn, Presta, Ganchev, Petrov & Collins 2016)[10], sind wegen der eingesetzten Technologie weniger einsteigerfreundlich und haben bisher noch keine Verbreitung in der deutschsprachigen Linguistik gefunden[11]. Es ist daher davon auszugehen, dass es hier in den nächsten Jahren noch zu gravierenden Veränderungen und Fortschritten kommen wird.

3. **Parameter beruhend auf einer Term-Dokumenten Matrix (TDM)**
   Bei einer Term-Dokumenten Matrix (TDM) handelt es sich um tabellarische Darstellungen von Texten. Hierbei repräsentiert jede Zeile ein Wort und jede Spalte, je nach Einteilung, einen gesamten Text oder einen Abschnitt. Bei den einzelnen Zellen handelt es sich um One-Hot-Vector Encodings oder gewichtete Frequenzen (Francis & Flynn 2010). Mithilfe dieser Tabellen kann die Ähnlichkeit verschiedener Wörter oder Texte miteinander verglichen werden. Somit kann Kohäsion, wenn zwei Sätze jeweils ein Document darstellen, oder semantische Nähe, wenn zwei Wörter miteinander verglichen werden, aufgezeigt und parametrisiert werden (siehe Kapitel 5.3.5).

Kurze Abschnitte können über die Webseite von Coh-Metrix [12] analysiert werden. Die Ausgabe erfolgt ohne Referenzwerte als HyperText Markup Language (HTML) Tabelle. Das Tool

[10] Vergleiche auch https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html (Stand: 19. August 2018)

[11] Eine Google Scholar Suche nach „Parsey McParseface" ergab nur einen deutschsprachigen Treffer (und bei diesem handelt es sich um eine wirtschaftswissenschaftliche Dissertation zu Requirements Engineering/abgerufen: 19. August 2018).

[12] http://tool.cohmetrix.com/ ; (Stand: 20. August 2018)

steht allerdings nicht als Download zur Verfügung oder kann per Schnittstelle angesprochen werden. Da ein Captcha verwendet wird, ist auch keine Abfrage per HTML Post möglich.

Für das oben beschriebene Vorgehen müssten die Rohdaten aus Coh-Metrix noch aufgearbeitet werden. Clusteringverfahren beheben das Kolliniaritätsproblem[13], während Standardisierungstransformationen eine höhere Gewichtung einzelner Variablen auf Grund ihrer höheren Zahlenwerte und Streuung verhindern. Auch wenn der Quellcode von Coh-Metrix unzugänglich ist, reicht die Dokumentation aus, um eine Vielzahl der korpuslinguistischen Variablen und Parameter zu reproduzieren (Gärtner 2014). Coh-Metrix wird ebenfalls dazu eingesetzt Schreibstile zu bestimmen (Stamatatos 2009), Sprachvariation und Kohäsion im Vergleich von gesprochener zu geschriebener Sprache zu analysieren (Louwerse, Mccarthy, Mcnamara & Graesser 2003) und Textgattungen zu unterscheiden (McCarthy, Briner, Rus & McNamara 2007).

Ein weiterer Anssatz um textuelle Komplexität zu ermitteln, wird vom Educational Testing Service (ETS) angeboten. Der ETS ist ein kommerzieller Anbieter von Sprachtests wie dem Test of Test of English as a Foreign Language (TOEFL) (Attali & Burstein 2006). Viele Universitäten verlangen für englischsprachige Studiengänge einen Nachweis für ausreichende Sprachkenntnisse. Unter anderem wird an der LUH für den konsekutiven Masterstudiengang Geodäsie und Geoinformatik ein solcher Sprachnachweis verlangt[14].

Das bisherige ETS-System bestand aus zwei von Menschen durchgeführten Bewertungen studentischer Essays. Die weltweite Verbreitung des TOEFL führt jedoch zu hunderttausenden zu korrigierenden Texten und daher zu einem entsprechenden Personalaufwand. Zur Kostenreduktion wurde ein automatisches Korrektursystem entwickelt. Der e-rater (Attali & Burstein 2006) ähnelt in seiner Grundidee Coh-Metrix. Komplexitätsmerkmale werden zum Feature Engineering genutzt und mittels einer linearen Regression als unabhängige Variablen zur Vorhersage der Essaynoten herangezogen. Neuere Publikationen deuten darauf hin, dass der Ansatz, linguistische Komplexitätsmerkmale zur Vorhersage von Essaynoten zu verwenden, kaum verändert wurde (Ramineni & Williamson 2018).

Da es sich beim ETS um einen kommerziellen Anbieter handelt, ist die entsprechende Software nicht frei verfügbar. Ebenso wie bei Coh-Metrix reicht die Dokumentation dennoch aus, um einen Großteil nachzuprogrammieren. Allerdings entsteht das Problem, ein ausreichend präzises Modell zu trainieren, um die Wahrscheinlichkeit, mit einer bestimmten Arbeit eine schriftliche Leistung zu bestehen, vorhersagen zu können. Der ETS nutzt eine große Masse an Texten als Trainingsdatensatz für seine Modelle. Soll der gleiche Ansatz verwendet werden, um die Erfolgswahrscheinlichkeit von schriftlichen Arbeiten vorherzusagen, muss eine ebenso große Textgrundlage herangezogen werden. Es ist zu vermuten, dass die an der LUH verfügbaren Dissertationen für diesen Einsatzzweck nicht in ausreichender Anzahl vorhanden sind. Um also den Einfluss von HanConc auf die Effektivität von Schreibberatung nachzu-

[13] Hoch kollineare Variablen wie Körpergröße, Körpergewicht und Schuhgröße beim Menschen verschlechtern die Genauigkeit von statistischen Modellen. Wenn aus inhaltlichen Gründen oder weil es sich um ein automatisiertes Verfahren handelt, nicht auf solche Variablenkombinationen verzichtet werden kann, können Clusteringverfahren diese Variablen zusammenfassen (Bortz 2010, Cameron & Trivedi 2005).

[14] §2 (6) der Ordnung über den Zugang und die Zulassung für den konsekutiven Masterstudiengang Geodäsie und Geoinformatik an der Gottfried Wilhelm LUH nach Verkündungsblatt der Gottfried Wilhelm LUH vom 29.05.2015

weisen, bedarf es eines größeren Forschungsprojektes. Ein solches wahrscheinlich nationales Forschungsprojekt übersteigt jedoch die Ziele dieser Arbeit.

2.2 Forschungsstand zur Schreibberatung

Die Literatur zum Thema Schreibberatung ist vielfältig. Sie soll in drei Kategorien eingeteilt werden: grundlegende theoretische Literatur, die vor der Gründung diverser Schreibzentren veröffentlicht wurde (etwa Kruse, Jakobs & Ruhmann 1999), Ratgeberliteratur von und für Schreibberatende und Erfahrungsberichte aus der Praxis von Schreibzentren. Auf eine Rezeption außereuropäischer Literatur soll mit Hinweis auf (Girgensohn & Peters 2012, 2) verzichtet werden, was

> [...] zum einen an der Sprachbarriere und zum anderen an den unterschiedlichen Entwicklungsständen in Bezug auf die Verbreitung von Schreibzentren sowie an den unterschiedlichen Universitätssystemen dies- und jenseits des Atlantiks [liegt], die es fraglich erscheinen lassen, ob diese Diskurse für die Arbeit in den europäischen Schreibzentren überhaupt relevant sind.

Zunächst soll auf zwei grundlegende Artikel zur theoretischen Fundierung von Schreibzentren hingewiesen werden:

Eines der ersten veröffentlichten Schreibmodelle stammt von John Hayes und Linda Flower aus dem Jahr (1980). Da es sich hierbei um eine psychologische Arbeit handelt, liegt der Fokus eher auf der Methodik, vorhandenes Fachwissen zu Papier zu bringen. Der eigentliche Prozess diese Ideensammlungen in eine fertige Arbeit zu transformieren tritt dabei in den Hintergrund. In ihrem Modell nehmen Hayes und Flower eine Dreiteilung des Schreibprozesses vor. Sowohl das Thema als auch das intendierte Publikum werden im ersten Teil des Schreibprozesses als Aufgabenstellung festgelegt. Das eigene Schreibprodukt wird dabei mehrfach mit diesen Vorgaben verglichen. Im zweiten Teil wird beschrieben, wie das Langzeitgedächtnis das notwendige Fachwissen, Kenntnisse über die Zielgruppe und bereits geschriebene Texte speichert und zur Verfügung stellt. Der dritte Teil des Modells umfasst das eigentliche Schreiben. Dieses wird noch einmal in vier Teile unterteilt: Der Text wird vorbereitet, geplant, geschrieben und danach überarbeitet. Auf einer Metaebene wird dieser Prozess etwa durch Schreibberatung unterstützt. Die einzelnen Teile des Modells gliedern sich jeweils in verschiedene Prozesse (Hayes & Flower 1980). Eine deutschsprachige Übersetzung dieses Modells kann in Grieshammer (2013) gefunden werden.

Der zweite Text von Hayes (1996) reorganisiert und erweitert das bisherige Modell. Weiterhin wird aus Sicht der Psychologie argumentiert. Jedoch wird das ursprüngliche Schreibmodell von drei auf zwei Teile reduziert. Hierbei werden der Prozess Wissen aus dem Langzeitgedächtnis abzurufen und der Schreibprozess miteinander verschmolzen. Dieser nun verschmolzene Prozess wird in neue Untereinheiten aufgeteilt. Hierbei wird zwischen Kurzzeit- und Langzeitgedächtnis, Motivation und Kognition unterschieden. Das eigentliche Schreiben

und Überarbeiten des Textes ist Teil des kognitiven Sub-Prozesses. Dabei wird der Einfluss von Schreibmedien wie Computern, Diktiergeräten, Stiften und Papier auf den kognitiven Schreibaufwand diskutiert. Wie Studierende bessere Texte schreiben können, wird jedoch nicht erörtert (Hayes 1996). Eine detailliertere Rezeption findet erneut in Grieshammer statt.

Im Folgenden wird deutschsprachige wissenschaftliche Ratgeberliteratur von und für Schreibberater_innen diskutiert:

Bei den wissenschaftlichen Texten (etwa (Ruhmann 1995, Kruse, Jakobs & Ruhmann 1999, Zegenhagen 2008, Ulmi, Bürki, Marti & Verhein-Jarren 2017)) sind zwei Zeitschriften und eine Autor_innenengruppe hervorzuheben. „Zeitschrift Schreiben" wird von der Pädagogischen Hochschule Zürich veröffentlicht und besteht seit 2006 [15]. Das „Journal der Schreibberatung" ist die zweite deutschsprachige regelmäßige Veröffentlichung zum Thema Schreibzentren und Schreibberatung. Dieses Journal wird an dieser Stelle erwähnt, da unter anderem mit Zegenhagen, Peters, Grieshammer und Liebetanz die Hauptakteurinnen[16] des deutschsprachigen Wissenschaftsbetriebs zum Thema Schreibzentren in diesem Journal veröffentlichen. Das Journal besteht seit 2010 und wird halbjährlich von einer privaten Gesellschaft bürgerlichen Rechts (GbR), bestehend aus den Herausgeberinnen[17], veröffentlicht. Beide Zeitschriften sind durch das wechselseitige Veröffentlichen eng miteinander verzahnt. Die Europa-Universität Viadrina Frankfurt (Oder) ist das wissenschaftliche Zentrum, aus dem die meisten Veröffentlichungen zur Schreibberatung stammen und in dem viele Gründer_innen weiterer Schreibzentren ausgebildet worden sind. Mit etwa Katrin Girgensohn, Simone Tschirpke, Franziska Liebetanz, Nora Peters und Anja Poloubotko hat eine Vielzahl an Autorinnen einen direkten Bezug zu dieser Universität.

Gemein ist vielen der Texte der Schreibberatungsratgeberliteratur, dass sie viele ihrer Ratschläge nur aus ihrer persönlichen Erfahrung heraus begründen oder auf andere Werke verweisen, die wiederum ebenso argumentieren. Die Begründungen bleiben dabei so oberflächlich, dass sie weder überprüft noch widerlegt werden können. Als Beispiel soll hier das Buchkapitel „Schreibberatung für internationale Studierende" von Sandra Ballweg dienen (2011). Sie schreibt mit Hinweis auf Gremmo (1995): „außerdem soll auch Unterstützung emotionaler Art gegeben werden" (Ballweg 2011, 124). Der Satz aus der Primärquelle, auf die sich diese These stützt, besagt, dass es die Aufgabe der Schreibberater_in sei:

> "[to give] psychological support, acting mostly as a 'benevolent outsider' who can help learners come to terms with their successes and failures."

Um diese These in die hier vorliegende Arbeit zu integrieren oder zu falsifizieren, müssten die einzelnen Teile parametrisierbar und überprüfbar sein. Es müsste geklärt werden, was eine Unterstützung „emotionaler Art" und ein „benevolent outsider" sein sollen und was mit „successes and failures" in diesem Zusammenhang gemeint sein soll. Ferner müsste nachgewiesen werden, dass eine solche Unterstützung die Anzahl an notwendigen Beratungssitzungen reduziert

[15] Alle Beiträge sind unter der Creative Commons Lizenz frei im Internet verfügbar; https://zeitschrift-schreiben.eu (Stand 26. August 2018)

[16] Es handelt sich hierbei ausschließlich um Frauen.

[17] Es handelt sich hierbei ausschließlich um Frauen.

oder die Bestehenswahrscheinlichkeit erhöht. Falls dies der Fall wäre, könnte eine emotionale Unterstützung Teil der oben genannten Formel werden. Andernfalls bindet ein Hauptfokus auf eine solche Vorgehensweise zeitliche Ressourcen der Schreibberater_in und verhindert gegebenenfalls die Arbeit an zielführenderen Facetten. Um einen empirischen Nachweis oder eine Falsifizierung zu erbringen, bedürfte es einer genaueren Beschreibung der Annahmen. Da jedoch eine Begründung ebenso wie eine Handlungsanweisung, wie denn diese Unterstützung aussehen soll, fehlt, kann eine Gegenthese genauso wie die These, dass eine emotionale Unterstützung irrelevant ist, nicht intersubjektiv und empirisch bewiesen werden.

Empirisch vorgehende Arbeiten wie Dittmann, Geneuss, Nennstiel & Quast (2003) bleiben in Bezug auf die in ihrer Arbeit untersuchte Zielgruppe fokussiert auf ihre akademischen Disziplinen. In der Studie werden 283 Studierende der Albert-Ludwigs Universität Freiburg zu ihren Schreibproblemen und ihren Anforderungen an eine Schreibberatung befragt. Bei der untersuchten Zielgruppe handelt es sich um geisteswissenschaftliche Studierende aus der Volkswirtschaftslehre, Linguistik, Literaturwissenschaft und Psychologie. Probleme durch das Verfassen von Arbeiten in der Zweit- oder Fremdsprache werden nicht beleuchtet. Die Diskrepanzen zwischen geisteswissenschaftlichen und einem mathematischen, ingenieurwissenschaftlichen, naturwissenschaftlichen oder technischen Studium in Bezug auf die Anforderungen an einen akademischen Text werden ebenso wenig beleuchtet.

Auf eine Diskussion von Erfahrungsberichten wird verzichtet, da diese nicht den Anspruch auf Wissenschaftlichkeit erheben. Es soll auf das „Journal der Schreibberatung" verwiesen werden, welches solche Berichte regelmäßig veröffentlicht.

Für das Schreibzentrum der LUH ergeben sich zwei besondere Schwierigkeiten, die eine Diskrepanz zu den Annahmen des oben skizzierten Forschungsstandes erzeugen. Während etwa Grieshammer (76, 2013) das Schreiben in einer Fremdsprache als Ausnahme sehen, so ist dies im oben genannten Schreibzentrum die Voraussetzung zur Teilnahme an Schreibberatungen. Studierende, die in ihrer Muttersprache schreiben, werden vom Zentrum für Schlüsselkompetenzen[18] betreut. Die zweite Schwierigkeit ergibt sich aus zu erwartenden Studierendengruppen. Auf Grund der geografischen Lage des Schreibzentrums und der Ausrichtung der LUH als ehemals technische Universität kann davon ausgegangen werden, dass die Mehrzahl der Studierenden mit Schreibberatungsbedarf ein mathematisch/technisches Studienfach belegt (siehe auch Kapitel 3 für eine tiefgreifendere Analyse der zu erwartenden Studierendengruppen). Würde die Schreibberatung in dieser Situation den Hinweisen aus „Zukunftsmodell Schreibberatung" folgen, wären Schreibberater_innen nur mit einem Duden und einem Fremdwörterbuch ausgestattet und mit ausländischen Studierenden konfrontiert, deren Texte sie auf Grund der Fachlichkeit und der gegebenenfalls schlechten sprachlichen Qualität nicht verstünden. Während Schreibberatungsliteratur entweder versucht die akademischen Inhalte für Schreibberater_innen zugänglich zu machen (Schroth-Wiechert 2011) oder auf die Bedürfnisse

[18]Das Schreibzentrum der LUH gehört zum Fachsprachenzentrum, an welchem etwa Französisch, Italienisch oder Business Englisch gelehrt wird, während das Zentrum für Schlüsselkompetenzen eine eigenständige Einrichtung an ebenjener Universität ist, an welchem verschiedene Zusatzqualifikationen (außer Fremdsprachen) erworben werden können.

von Studierenden einzugehen, die nicht in ihrer Erstsprache schreiben (Myers 2003, Nakamaru 2010, Matsuda 1999), ergibt sich eine Forschungslücke zu technisch unterstützter Schreibberatung in einer Fremdsprache bei fachlich inhaltlicher Diskrepanz zu Schreibberatenden. Ziel dieser Arbeit ist es dabei technische Hilfsmittel zur Verfügung zu stellen und gleichzeitig pädagogische Freiheit bei der Benutzung zu ermöglichen.

Kapitel 3

Zielgruppenanalyse

Das Schreibzentrum der Leibniz Universität Hannover (LUH) und diese Arbeit orientieren sich an der Annahme, dass es ausreichend viele Studierende gibt, die in einer Zweit- oder Fremdsprache schriftliche Arbeiten schreiben müssen und dabei Unterstützung benötigen. Dieses Kapitel wertet die zur Verfügung stehenden Zahlen des Immatrikulationsamtes, der Studienordnungen und Vorlesungsverzeichnisse mit Stand vom Wintersemester 2015/2016 aus, um diese Annahme zu bestätigen. Kapitel 3.1 zeigt die Verteilung internationaler Studierender im Zeitverlauf und nach Herkunftsländern und Fakultäten auf Basis der Daten des Immatrikulationsamtes. Kapitel 3.1.1 erläutert die Ermittlung der Schreibaufwände einzelner Fakultäten und Kapitel 3.1.2 die Verteilung dieser Aufwände auf die einzelnen Gebäude der LUH. Auf Basis dieser Daten wird die Zielgruppe des Schreibzentrums in Bezug auf die Anzahl an Studierenden, ihren Schreibaufwand, ihre Herkunftsländer und ihre Fakultätszugehörigkeit beschrieben. Kapitel 3.2 wertet die Protokolle eine Schreibberaters aus, um zu überprüfen, ob sich die in Kapitel 3.1 und Kapitel 3.1.1 aufgestellten Theorien bewahrheiten. Abschließend wird mit einer Umfrage in Kapitel 3.3 betrachtet, ob sich die bisherigen Erkenntnisse mit den Erfahrungen in anderen Schreibzentren decken.

3.1 Internationale Studierende

Im folgenden Kapitel soll die Gruppe der internationalen Studierenden an der LUH als mögliche Zielgruppe für Schreibberatungen näher betrachtet werden. Hierzu soll von folgenden Annahmen ausgegangen werden:

- Internationale Studierende bilden kulturell und sprachlich heterogene Gruppen
- Unterschiedliche Faktoren beeinflussen die einzelnen Gruppengrößen:
  - Bevölkerungsgröße und -struktur der Herkunftsländer
  - Wirtschaftskraft der Herkunftsländer
  - Finanzielle Förderung durch Herkunftsländer
  - Politische Entwicklungen in den Herkunftsländern

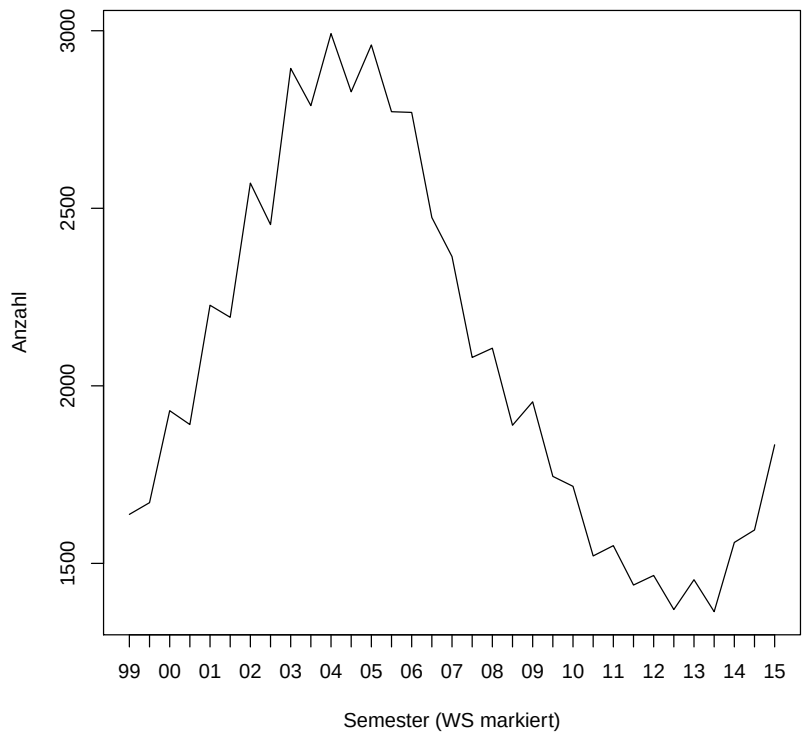


Abbildung 3.1: Gesamtanzahl internationaler Studierender pro Semester (Wintersemester markiert) an der LUH

- Kulturell begründet wird einzelnen Studiengängen und den damit zusammenhängenden Berufsbildern unterschiedlich viel Prestige zugewiesen
- Die Vorbereitung auf das deutsche Universitätssystem unterscheidet sich je nach Herkunftsbildungssystem

Einige dieser Faktoren können kaum so operationalisiert werden, dass sie mit den zur Verfügung stehenden Daten oder anderen öffentlich verfügbaren Quellen erfasst werden könnten. Während die Gruppengröße aus den Daten des Immatrikulations- und Prüfungsamtes abgelesen werden kann, ist es kaum möglich den Einfluss verschiedener anderer Faktoren zu erfassen. Zum Beispiel stieg die Zahl syrischer Studierender zwar seit dem Beginn des Bürgerkriegs, jedoch begann dieses Wachstum bereits 2010 und geht kaum über die Schwankungen anderer Studierendengruppen hinaus. Deshalb sollen in diesem Kapitel auf Basis der Daten des Immatrikulationsamtes und des Prüfungsamtes die vielversprechendsten Zielgruppen für Schreibberatungen ermittelt werden.

Abbildung 3.1 zeigt die Anzahl internationaler Studierender zwischen dem Wintersemester 1999/2000 und dem Wintersemester 2015/2016. Die Gesamtanzahl enthält beurlaubte Studierende, jedoch keine Promotions-, Austausch- oder Musikstudent_innen oder Teilnehmer_innen von Deutschkursen.

Insgesamt ist eine wellenförmige Entwicklung der Studierendenzahlen zu beobachten. Im Wintersemester 2004/2005 gab es mit einer Anzahl von 2.992 mehr als doppelt so viele interna-

tionale Studierende wie im Sommersemester 2013 (1.364). Der Mittelwert über den gesamten Zeitraum liegt bei etwa 2.000.

Auf den ersten Blick könnte diese Entwicklung mit der Internationalisierungsstrategie der LUH und den Studiengebühren des Landes Niedersachsen begründet werden, denn entsprechend der Internationalisierungsstrategie soll die LUH attraktiver für ausländische Studierende werden. Außerdem könnten die Studiengebühren zwischen dem Wintersemester 2005/2006 und dem Wintersemester 2014/2015 zu einem Einbruch geführt haben[1].

Abbildung 3.2 zeigt eine Aufschlüsselung von Abbildung 3.1 nach Herkunftsland und Fakultät. Bei den Herkunftsländern stechen vor allem die Volksrepublik China (obere schwarze Linie), die Russische Föderation und Polen hervor. Diese drei Länder bestreiten den Hauptanteil der ausländischen Studierenden an der LUH. Ebenso wie bei den anderen Herkunftsländern erreicht ihre Anzahl 2005 ihren Höhepunkt und sinkt danach. Erst 2012 steigen die Zahlen wieder. Auffällig ist vor allem das Fehlen sämtlicher Anrainerstaaten Deutschlands außer Polen unter den 10 häufigsten Herkunftsländern.

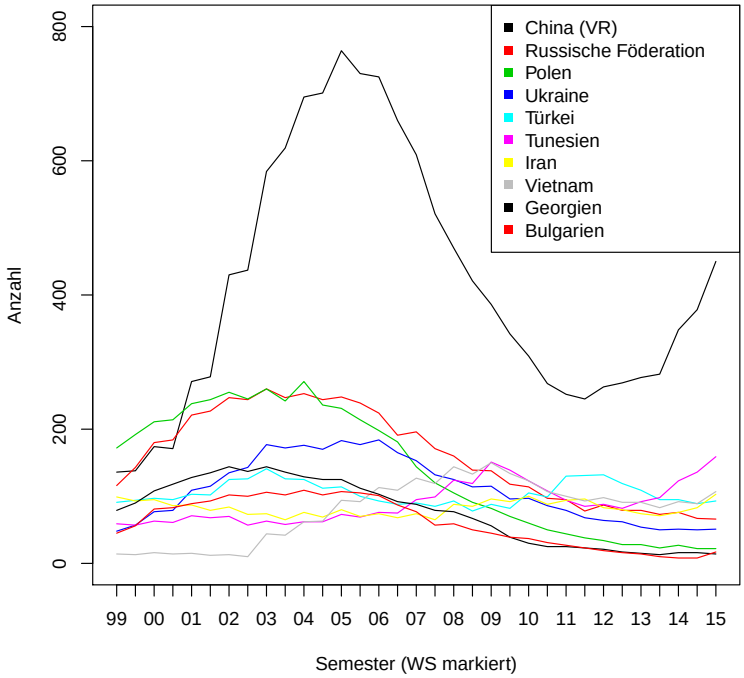
In Bezug auf die absolute Anzahl[2] an ausländischen Studierenden stechen drei Fakultäten besonders hervor. Die höchste absolute Anzahl an ausländischen Studierenden findet sich an der philosophischen Fakultät FPhi. Hier dominieren vor allem Polen, die Russische Föderation, Georgien und die Volksrepublik China die Liste der häufigsten Herkunftsländer. Die zweit häufigste Gruppe besteht aus ausländischen Studierenden an der Fakultät für Elektrotechnik und Informatik (FElt) und der Fakultät für Maschinenbau (FMas). An beiden Fakultäten stammen die größten Gruppen an Studierenden aus der Volksrepublik China, Tunesien und Vietnam. Vor allem der Anstieg an chinesischen Studierenden am Anfang des Jahrtausends führt zu den ansteigenden Zahlen bei diesen Fakultäten, während der Abfall ab 2005 von den beiden letzteren Gruppen teilweise aufgefangen wird. Die übrigen Fakultäten weisen eine deutlich niedrigere Zahl an ausländischen Studierenden auf.

Wie Abbildung 3.2(a) bereits zeigt, ist die Verteilung der Herkunftsländer nicht gleichverteilt. Die farbliche Markierung der Herkunftsländer der Weltkarte (Abbildung 3.3) zeigt die Anzahl der Studierenden aus diesen Ländern im Wintersemester 2015/2016. Es sind fünf geografische Cluster zu erkennen:

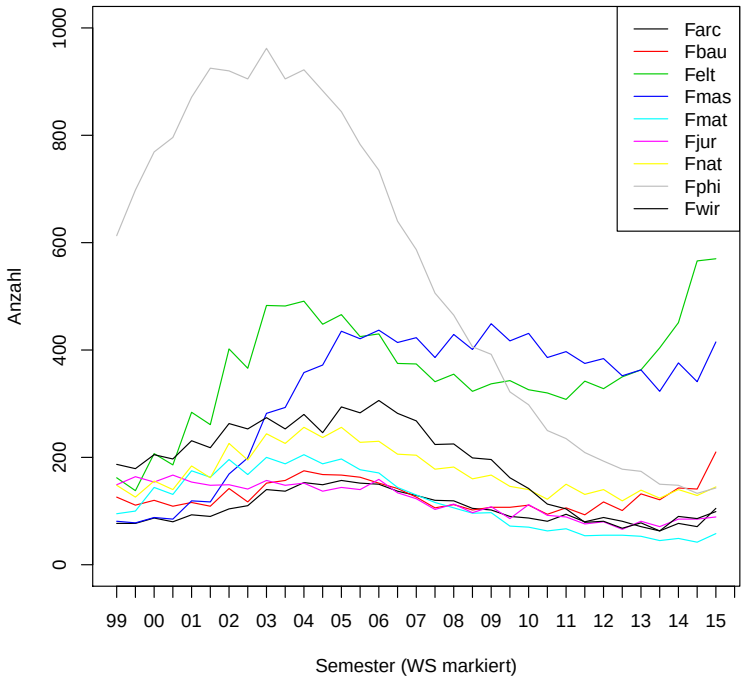
- Russland und die Ukraine
- Tunesien
- Der Nahe Osten, d.h. Türkei, Syrien und Iran
- Süd-Ost Asien
- Volksrepublik China

[1] Die Studiengebühren wurden tatsächlich erst ein Jahr später eingeführt. Die Ankündigung erfolgte allerdings schon zu besagtem Zeitpunkt.

[2] Diese Analyse dient unter anderem dazu, zu ergründen, welche ausländischen Studierenden am Schreibzentrum der LUH erwartet werden können. Daher ist das Verhältnis der absoluten Anzahl an ausländischen Studierenden zwischen den Fakultäten wichtiger als das quantitative Verhältnis von ausländischen zu inländischen Studierenden innerhalb einer Fakultät.



(a) Internationale Studierende pro Semester nach Herkunftsland



(b) Internationale Studierende pro Semester und Fakultät

Abbildung 3.2: Abbildung 3.1 aufgeschlüsselt nach Herkunftsland und Fakultät

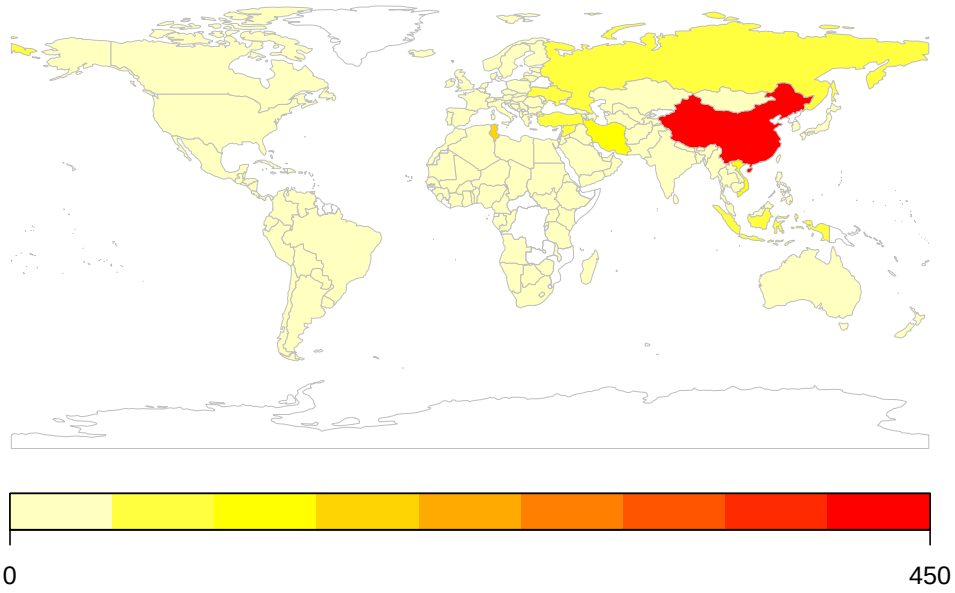


Abbildung 3.3: Anzahl internationaler Studierender nach Herkunftsland im Wintersemester 2015/2016

An der Farbgebung ist zu erkennen, dass vor allem aus Tunesien und China viele Studierende kommen. Die absoluten Zahlen zeigen, dass aus China etwa drei mal so viele Studierende kommen wie aus Tunesien und vier ein halb mal so viele wie aus Vietnam (Tabelle 3.1). Auch hier zeigt sich, dass die Bevölkerungsgröße der Herkunftsländer nicht unbedingt ausschlaggebend für die Gruppengröße ist. So kommen trotz einer ähnlichen Bevölkerungsgröße nur etwa 10% so viele Inder_innen wie Chines_innen nach Hannover.

Tabelle 3.1: 20 häufigste Herkunftsländer internationaler Studierender im Wintersemester 2015/2016

Land	Anzahl	Land	Anzahl
Volksrepublik China	450	Indien	36
Tunesien	159	Kamerun	28
Vietnam	107	Palästin. Gebiete	24
Iran	103	Polen	22
Türkei	93	Kolumbien	18
Syrien	85	Bangladesch	17
Indonesien	85	Bulgarien	17
Russische Föderation	66	Albanien	16
Ukraine	51	USA	15
Libanon	43	Georgien	14

Ebenso wie die Verteilung der Herkunftsländer auf die Fakultäten ist auch die Erfolgswahrscheinlichkeit, ein Studium zu beenden, unterschiedlich. Abbildung 3.4 zeigt für ausgewählte

Kombinationen aus Fakultät und Herkunftsland die Anzahl an Studierenden zu Abschlüssen. Da die Abschlüsse zeitversetzt zu der Gesamtanzahl an Studierenden gesehen werden müssen, wird die Anzahl an Studierenden konstant gehalten und die Anzahl an Abschlüssen so versetzt, dass der Korrelationskoeffizient optimiert wird. Dieser Zeitverzug (Lag) beträgt zwischen zwei und sieben Semestern. In allen Fällen sollten sich beide Kennzahlen stabilisieren und parallel zueinander verlaufen. Jedoch zeigen Abbildungen 3.4(b-d), dass nur wenige chinesische Studierende ihren Abschluss schaffen. Im Gegensatz dazu zeigen die Daten für iranische Elektro- und Informationstechnikstudierende eine deutlich höhere Erfolgsquote. Leider erlauben die Datenschutzregelungen der LUH keine genaueren Analysen.

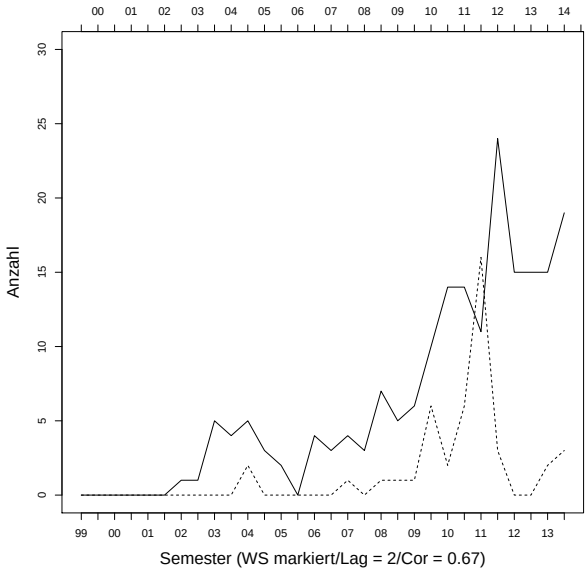
In diesem Kapitel wurde herausgearbeitet, dass geografische Cluster bei den Herkunftsländern der ausländischen Studierenden zu finden sind. Ebenso verteilt sich die Anzahl an ausländischen Studierenden ungleichmäßig auf die einzelnen Fakultäten mit einem massiven Überhang an der Philosophischen Fakultät und den ingenieurwissenschaftlichen Fakultäten. Die Notwendigkeit einer Unterstützung der Studierenden ergibt sich aus den niedrigen Abschlusszahlen bei einigen Kombinationen aus Herkunftsland und Fakultät bzw. Fachgebiet. Auf Grund des Datenschutzes ist es leider nicht möglich, nachzuvollziehen, zu welchem Zeitpunkt in ihrem Studium die Studierenden abbrechen. Vor allem Abbrüche in höheren Semestern würden auf Schwierigkeiten bei schriftlichen Ausarbeitungen hindeuten, da diese erst im letzten Abschnitt des Studiums geschrieben werden (siehe folgendes Kapitel) und bis dahin bereits eine Gewöhnung an die Anforderungen des Faches stattgefunden haben sollte. Vor allem zur Kapazitätssteuerung der Fakultäten sollte hier intensiver analysiert werden, um zu verhindern, dass Studierende in niedrigen Semestern Kapazitäten belegen und dann auf Grund fehlender Schreibberatung und Unterstützung in den letzten Semestern scheitern.

3.1.1 Schreibaufwand internationaler Studierender nach Fakultät

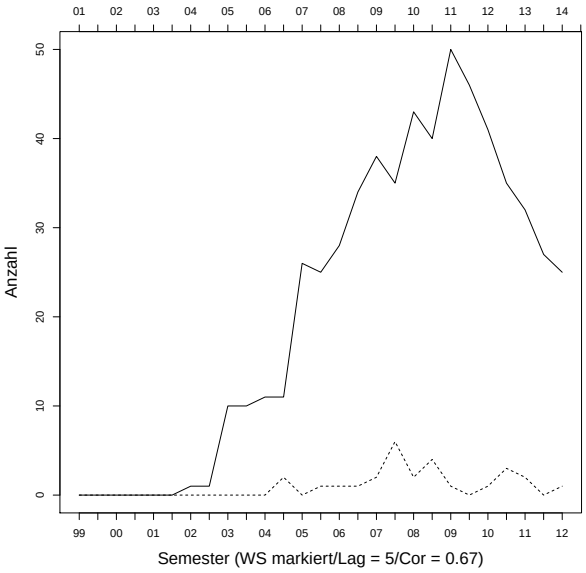
Im letzten Kapitel wurden bereits die größten Studierendengruppen anhand ihrer Herkunftsländer, Studienrichtungen und Erfolgswahrscheinlichkeiten, ihr Studium auch zu beenden, analysiert. Jedoch ist der Schreibaufwand, um ein Studium erfolgreich zu absolvieren, je nach Studienfach unterschiedlich. Geisteswissenschaftliche Studienfächer haben tendenziell einen höheren Schreibaufwand als die sogenannten MINT-Fächer (Mathematik, Informatik, Naturwissenschaften, Technik). Um diese These zu überprüfen, sollen die Daten aus den Prüfungsordnungen der LUH ausgewertet werden. Ziel dieser Analyse ist es, die Grundlage für die Anpassbarkeit vom Hannover Concordancer (HanConc) an seine Benutzer_innen zu ermöglichen. Dies soll einerseits inhaltlich über verschiedene fakultätsspezifische Subkorpora geschehen und andererseits über die Anpassung der Funktionen und des Graphical User Interface (GUI).

Um abzuschätzen, wie viele Studierende aus bestimmten Herkunftsländern ein Schreibzentrum in Anspruch nehmen würden, ist nicht nur ihre absolute Anzahl in Betracht zu ziehen, sondern auch Informationen zum Schreibaufwand ihrer Fächer und der lokalen Erreichbarkeit des Schreibzentrums.

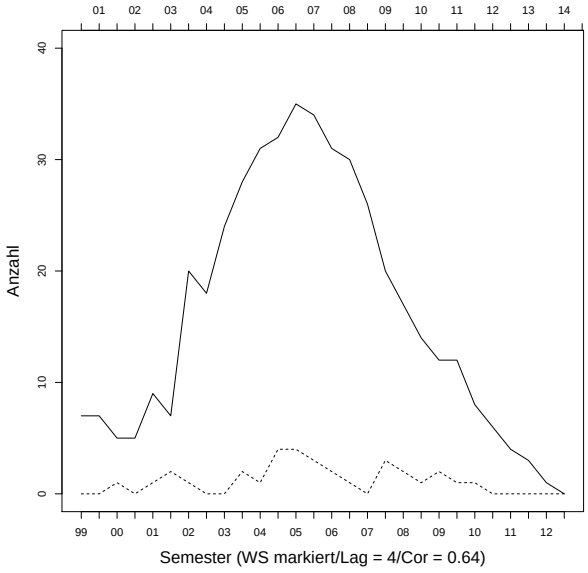
Bei gleicher absoluter Anzahl an internationalen Studierenden würden mehr Schreibbe-



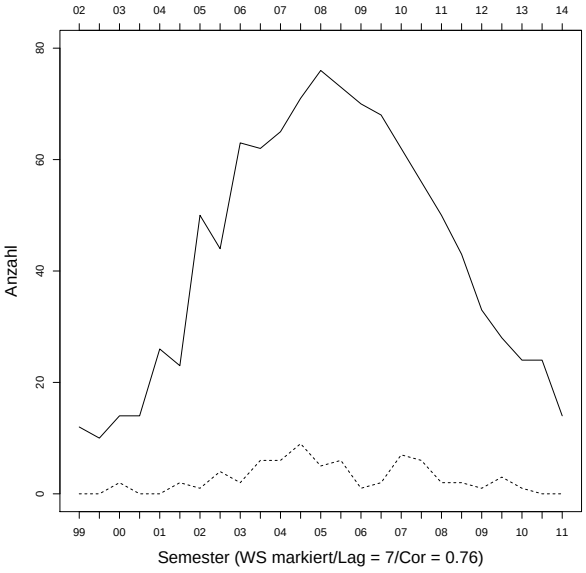
(a) FElt/Iran



(b) FMas/Vietnam



(c) FBau/China



(d) FElt/China

Abbildung 3.4: Ausgewählte Kombinationen aus Fakultät und Herkunftsland mit jeweils der Anzahl an eingeschriebenen Studierenden pro Semester und optimal zeitlich verschoben die Anzahl an Abschlüssen

ratungen von Studierenden aus der Fakultät in Anspruch genommen werden, an der sie am meisten im Verlauf ihres Studiums schreiben müssen. Außerdem ist davon auszugehen, dass Schreibberatungen tendenziell für solche Arbeiten in Anspruch genommen werden, bei denen der Aufwand, zum Schreibzentrum zu gelangen und die Zeit, welche in den Schreibberatungen verbracht wird, in einem günstigen Verhältnis zum erwarteten Nutzen stehen. Das heißt, dass der Druck bei einer schriftlichen Leistung, deren Benotung einen nicht unerheblichen Teil der Abschlussnote ausmacht, höher ist, als bei einer unbenoteten Leistung.

Der zeitliche Aufwand und die Gewichtung der einzelnen Arbeiten im Verhältnis zur Gesamtbewertung kann in den jeweiligen Prüfungsordnungen anhand der Leistungspunkte, vergeben nach dem European-Credit-Transfer System (ECTS), abgelesen werden. §2 der Bachelorprüfungsordnungen bzw. §8 der Masterprüfungsordnungen an der LUH regelt die Dauer und Gliederung des Studiums in ECTS-Leistungspunkten. Es werden je 30 ECTS-Leistungspunkte pro Semester zu je 30 Zeitstunden angesetzt (Europäische Kommision 2015). Die Gesamtnote errechnet sich aus dem gewichteten Mittel der Einzelleistungen.

Um also den relativen Schreibaufwand eines Studienganges zu ermitteln, können die Leistungspunkte aus benoteten schriftlichen Leistungen (Prüfungsleistungen) addiert und durch die Gesamtzahl der Leistungspunkte des Studienganges dividiert werden[3]. Formel 3.1 beschreibt diesen Zusammenhang:

$$E(SL|Studiengang) = \frac{\sum CP_{SL|Studiengang}}{CP_{Studiengang}} \cdot 100 \tag{3.1}$$

wobei SL = schriftliche Leistungen und CP = Credit Points.

Wie schon in den vorherigen Kapiteln sollen die einzelnen Studiengänge zu Fakultäten zusammengefasst werden. Hierzu werden die CP aus schriftlichen Leistungen der einzelnen Studiengänge einer Fakultät addiert und durch die Gesamtzahl der CP der Fakultät dividiert (siehe Formel 3.2)

$$E(SL|\textit{Fakultät}) = \frac{\sum CP_{SL|Studiengang \cap \textit{Fakultät}}}{\sum CP_{Studiengang \cap \textit{Fakultät}}} \cdot 100 \tag{3.2}$$

wobei SL = schriftliche Leistungen und CP = Credit Points.

Mit Hilfe dieser Formeln lässt sich der Anteil schriftlicher Leistungen einer Fakultät abschätzen. Einschränkend ist allerdings festzuhalten, dass sowohl der fächerübergreifende Bachelor und der Master of Education der Philosophischen Fakultät als auch die Abschlüsse der juristischen Fakultät nicht enthalten sind. Der fächerübergreifende Bachelor ist an der philosophischen und der naturwissenschaftlichen Fakultät, der Fakultät für Mathematik und Physik der LUH sowie der Hochschule für Musik, Theater und Medien Hannover angesiedelt. Studierende dieses Bachelorstudiengangs belegen zwei Studiengänge an diesen Fakultäten bzw. Hochschulen. Somit ist eine Zuordnung zu einer Fakultät nicht möglich. Gleiches gilt für den Abschluss Master of Education. Die juristische Fakultät ist nicht im Bachelor/Master System organisiert,

[3]Credit Points (CP), Leistungspunkte und ECTS-Leistungspunkte beschreiben das gleiche Konzept. In dieser Arbeit wird das englische Wort Credit Points (CP) verwendet.

sondern vergibt Staatsexamen. Eine Vergleichbarkeit ist somit nicht mehr gegeben und diese Fakultät wird entsprechend nicht in dieser Arbeit betrachtet.

Tabelle 3.2 zeigt den absoluten und relativen Schreibaufwand je Fakultät. Auffällig ist die Variation der einzelnen Fakultäten. Während bei der philosophischen Fakultät über 80% der CP mit schriftlichen Leistungen erbracht werden, sind es bei den Fakultäten für Elektrotechnik, Maschinenbau, Mathematik und Wirtschaft jeweils nur zwischen 13% und 15%.

Tabelle 3.2: Schreibaufwand durch Prüfungsleistungen in CP pro Fakultät

Fakultät	CP Aufwand	CP Gesamt	Anteil in %
FArc	212	540	39,259
FBau	337	1.500	22,467
FElt	166	1.200	13,833
FMas	160	1.020	15,686
FMat	163	1.020	15,980
FNat	632	2.940	21,497
FPhi	1.226	1.500	81,733
FWir	80	600	13,333

Anhand von Tabelle 3.2 wäre zu erwarten, dass Studierende der Philosophischen Fakultät eine fünf mal höhere Wahrscheinlichkeit haben, eine Schreibberatung in Anspruch zu nehmen. Die internen Daten der Schreibberater_innen des multilingualen Schreibzentrums der LUH widersprechen jedoch dieser Annahme: Wie Kapitel 3.2 zeigt, nutzen vor allem Studierende ingenieurwissenschaftlicher Studiengänge die Schreibberatungsangebote.

Zu einer genaueren Analyse sollen noch zwei weitere Informationsquellen hinzugefügt werden. Aus den Studierendendaten der LUH lässt sich die Anzahl internationaler Studierender je Fakultät ablesen (siehe Kapitel 3.1). Es wird überprüft, ob ein Ungleichgewicht in der absoluten Anzahl an internationalen Studierenden das Ungleichgewicht des Schreibaufwands ausgleichen kann. Weil außerdem davon auszugehen ist, dass Studierende, die in der Nähe des Schreibzentrums den Großteil ihres Studiums verbringen, eher das Schreibzentrum aufsuchen, sollen die potentiellen Aufenthaltsorte anhand der Vorlesungsverzeichnisse geschätzt werden.

In den Vorlesungsverzeichnissen der unterschiedlichen Fakultäten bzw. Institute sind alle Veranstaltungen der jeweiligen Einrichtungen vermerkt. Außerdem wird jeweils der Raum und das Gebäude angegeben. Alle Gebäude der LUH haben eine vierstellige Nummer, wobei die ersten zwei Zahlen für den Campus stehen. Abbildung 3.5 zeigt wie aus den kommentierten Vorlesungsverzeichnissen (KVV) die Anzahl an Veranstaltungen pro Fakultät und Campus extrahiert werden kann.

Die so erhaltenen Zahlen sind allerdings nur Schätzungen, da die Kurs- bzw. Vorlesungsgröße und -auslastung, sowie Musterstudienpläne nicht berücksichtigt werden können. Eine genauere Analyse wäre mit den Daten aus der universitätsinternen Veranstaltungsverwaltungssoftware und personalisierten Daten aus dem Zahlenspiegel möglich. Leider wurden diese Daten für diese Arbeit nicht zur Verfügung gestellt.

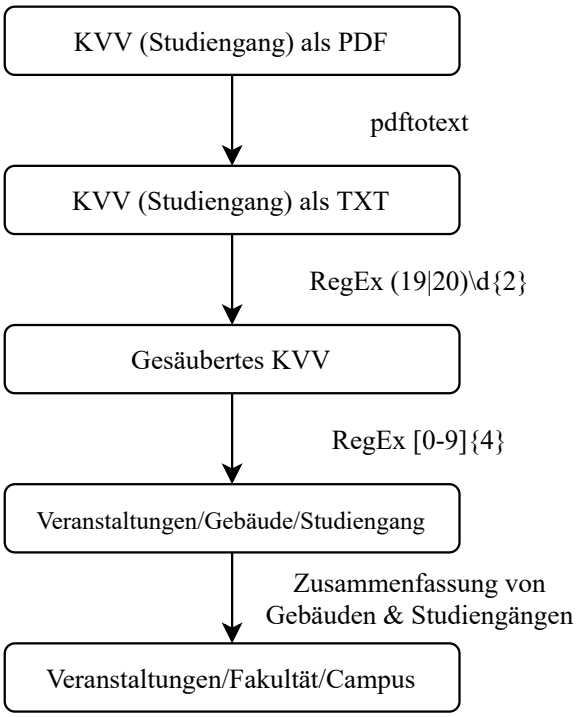


Abbildung 3.5: Programmablaufplan zur Ermittlung der zu erwartenden Anzahl internationaler Studierender pro Campus und Fakultät auf Basis der kommentierten Vorlesungsverzeichnisse für das Wintersemester 2015/2016

Für eine potentielle Schreibberatung kommen also nur Studierende in Frage, die auch eine Arbeit schreiben und an einem Campus in der Nähe des Schreibzentrums studieren. Der Erwartungswert für den Umfang und die Anzahl an schriftlichen Leistungen soll in Form von CP pro Campus und Fakultät unter Berücksichtigung der Anzahl internationaler Studierender angegeben werden. Als Formel ausgedrückt ergibt dies:

$$E(CP|Campus \cap Fakultät) = N_{IST|Fakultät} \cdot \frac{E(VA|Campus \cap Fakultät)}{100} \cdot \frac{E(SL|Fakultät)}{100} \tag{3.3}$$

wobei CP = Credit Points, VA = Veranstaltungsanzahl und SL = schriftliche Leistungen.

Abbildung 3.6 zeigt die Anzahl internationaler Studierender je Fakultät und Campus anhand von verschieden großen Tortendiagrammen, während die Farbverteilung den Anteil der einzelnen Fakultäten daran wiedergibt. Die Zahlen unter den Diagrammen entsprechen dem Campus. Campus 63 (Bismarckstraße), 73 (Ruthe), 81 (Garbsen) und 89 (Garbsen) sind auf Grund ihrer Entfernung zum Innenstadtcampus nicht auf den Karten verzeichnet[4].

In der Karte sind zwei Schwerpunkte zu erkennen: Campus 11 und Campus 34 mit 417 bzw. 360 zu erwartenden internationalen Studierenden (siehe auch Abbildung 3.5). Bei beiden Standorten stellen Studierende der Ingenieurwissenschaften die Mehrheit. Die hohen Zahlen ergeben sich aus der Akkumulation der einzelnen ingenieurwissenschaftlichen Fakultäten auf wenige Standorte und der gleichzeitig hohen Streuung der Veranstaltungen der philosophischen

[4]Es ist an dieser Stelle zu bedenken, dass die Zahlen aus dem Jahr 2016 stammen. Zu dieser Zeit war der Maschinenbaucampus in Garbsen noch nicht fertiggestellt.

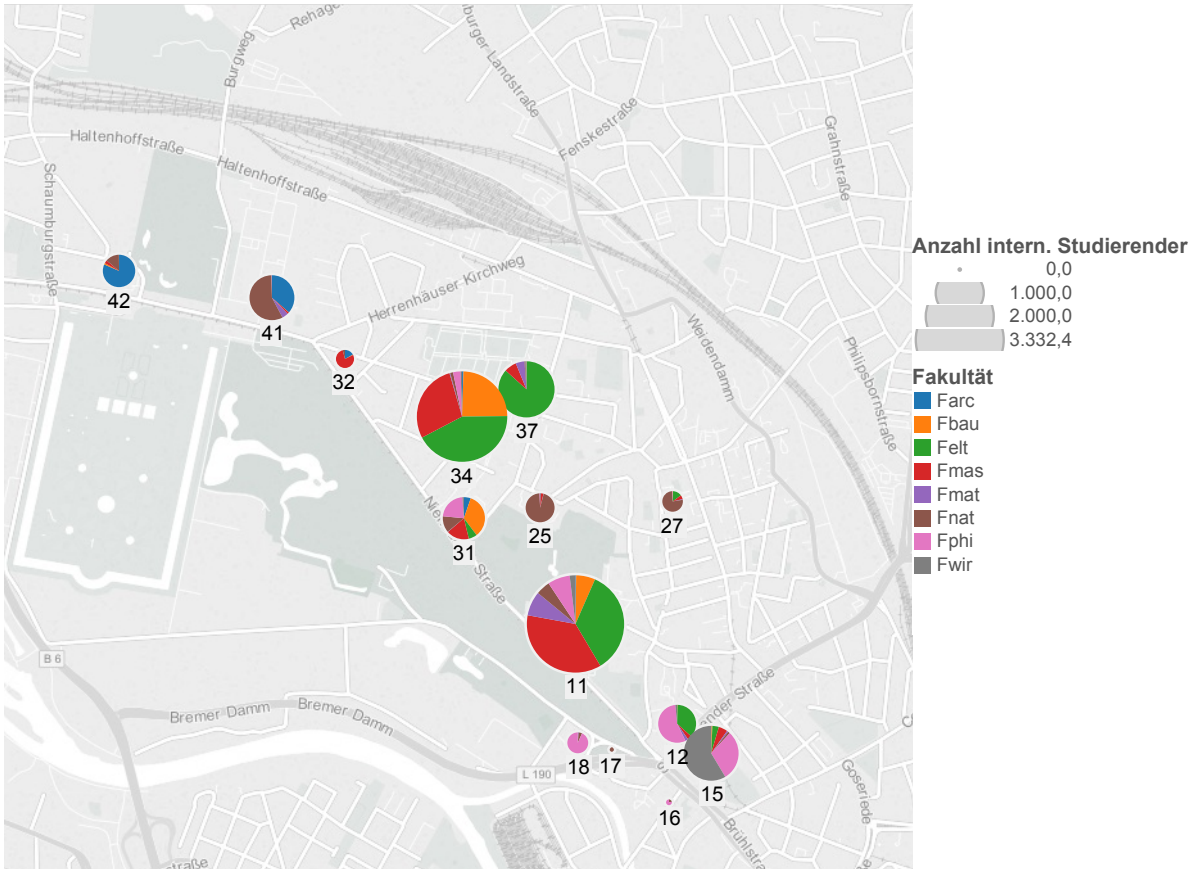


Abbildung 3.6: Erwartete Anzahl internationaler Studierender pro Campus und Fakultät

Fakultät (siehe auch Kapitel 3.1).

Abbildung 3.7 zeigt die zu erwartenden CP pro Campus und Fakultät. Auch hier korrespondiert die Größe der Tortendiagramme mit der erwarteten Anzahl an CP, während die Aufteilung innerhalb des Diagramms dem jeweiligen Anteil der einzelnen Fakultäten entspricht. Im Vergleich zu Abbildung 3.6 fällt eine Erhöhung der Anzahl an CP der Philosophischen Fakultät auf. Der deutlich höhere Schreibaufwand gleicht die geringere absolute Anzahl an internationalen Studierenden aus. Dennoch verteilt sich ein großer Teil der CP in der Nähe des Schreibzentrums (Gebäude 3110 auf Campus 31) auf ingenieurwissenschaftliche Studiengänge. Dies wird besonders in Abbildung 3.8 deutlich.

Die Abbildungen 3.8(a) und 3.8(b) zeigen jeweils die zu erwartenden CP aus schriftlichen Leistungen von ausländischen Studierenden pro Campus in Abhängigkeit vom Abstand zum Schreibzentrum. In Abbildung 3.8(b) sind zusätzlich Trendlinien für jede Fakultät und für die kombinierten ingenieurwissenschaftlichen Fakultäten vermerkt. Auffällig ist hierbei, dass die drei ingenieurwissenschaftlichen Fakultäten (FBau, FElt und FMas) ihre höchste CP Konzentration im Bereich von 500 bis 750m um das Schreibzentrum herum haben, während die Konzentration bei der FArc und der FPhi ihren Höhepunkt erst bei 1.300 bis 1.600m erreicht. Durch die schwarze Trendlinie wird deutlich, dass auf den ersten 500m von den drei ingenieurwissenschaftlichen Fakultäten eine etwa drei mal höhere CP-Konzentration erreicht wird als von den anderen Fakultäten.

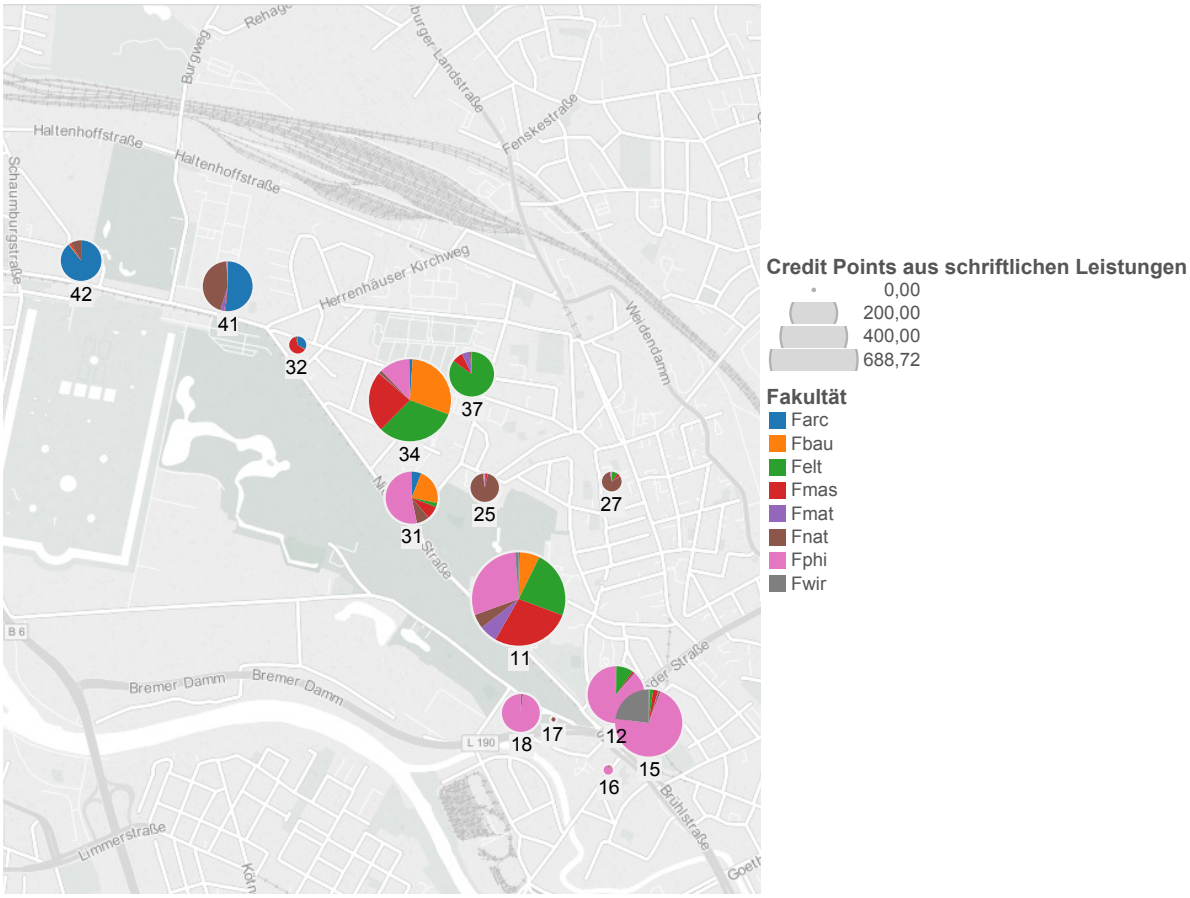
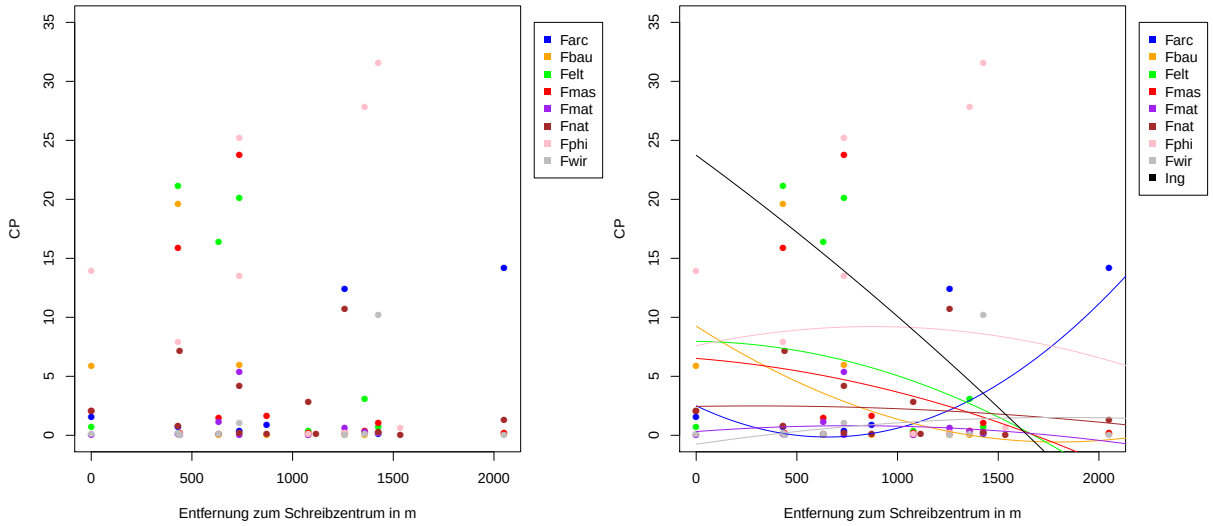


Abbildung 3.7: Erwartete CP aus schriftlichen Leistungen pro Campus und Fakultät



(a) Ohne Trendlinien (b) Mit Trendlinien

Abbildung 3.8: CP pro Campus in Abhängigkeit von der Entfernung zum Schreibzentrum in Metern

Sollte die Vermutung zutreffen, dass eine größere Distanz zum Schreibzentrum einen Einfluss auf die Wahrscheinlichkeit, eine Schreibberatung in Anspruch zu nehmen, hat, dann sollte sich dies in den Protokollen des Schreibzentrums widerspiegeln.

3.1.2 Geographic Profiling

Die Analysen des vorherigen Kapitels sind rein deskriptiv. Sie zeigen, dass ein Großteil der zu erwartenden schriftlichen Leistungen ausländischer Studierender an den ingenieurwissenschaftlichen Fakultäten bzw. in deren Gebäuden erbracht wird. Um die optimale Position, d.h. den Ort, von dem aus die meisten Studierenden bzw. CP erreicht werden können, zu bestimmen, soll ein anderes Verfahren gewählt werden. Durch die folgende Analyse wird gezeigt, dass der aktuelle Standort des Multilingualen Schreibzentrums im mathematischen Sinne optimal ist, um eine möglichst hohe Abdeckung zu erreichen.

Das Geographic Profiling (GP) ist ein statistisches Verfahren, das in der Kriminologie und Epidemiologie eingesetzt wird, um den potentiellen Aufenthaltsort von Kriminellen bzw. die Quelle einer Epidemie aufzuspüren (Verity, Stevenson, Rossmo, Nichols & Comber 2014). In dieser Arbeit soll auf Basis der Analyse aus dem vorherigen Kapitel überprüft werden, wo der Ursprung oder die Ursprünge der CP aus schriftlichen Leistungen liegen und sich somit der optimale Standort für ein Schreibzentrum befindet.

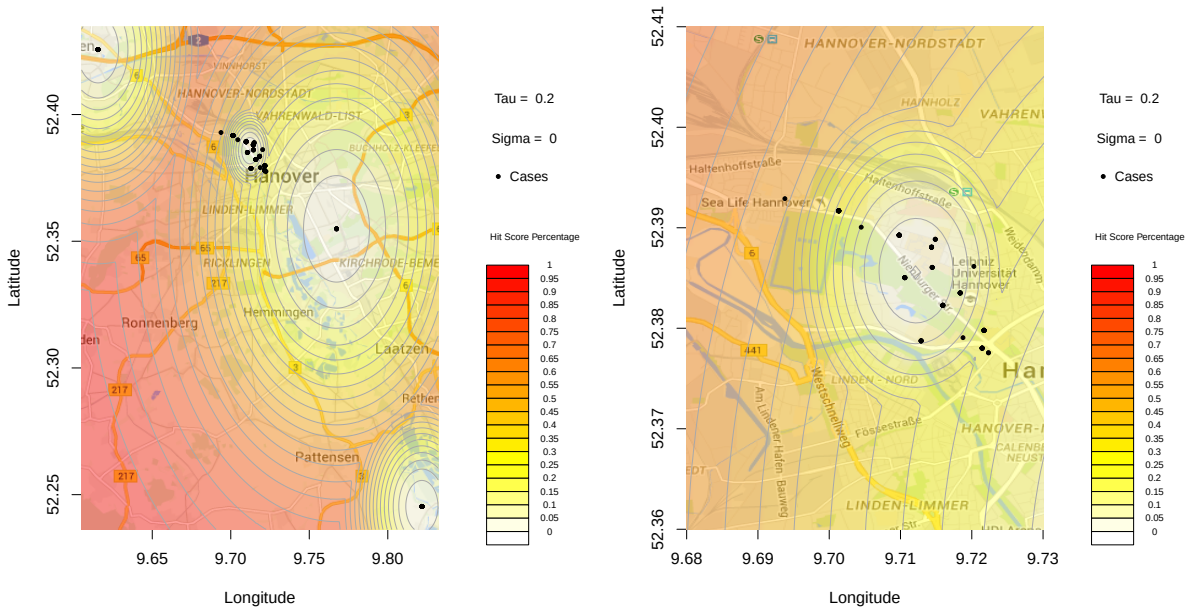
Im Folgenden wird ein Dirichlet Process Mixture Model (DPM) in der Umsetzung als Rgeoprofile R Paket genutzt (Spaulding & Morris 2021), um die Distanz einer potentiellen Quelle zu den einzelnen Standorten zu modellieren und schlussendlich zu clustern. Die Analyse soll zwei mal vollzogen werden: In einem ersten Durchlauf werden die geografischen Eigenschaften der einzelnen Standorte betrachtet und in einem zweiten werden zusätzlich die Ergebnisse aus Kapitel 3.1.1 integriert[5].

Zunächst werden jeweils die Mittelpunkte der einzelnen Standorte als ein Incident gewählt und mittels DPM geclustert. Abbildung 3.9 zeigt die Ergebnisse im Überblick und als Detailansicht. Bei den schwarzen Punkten handelt es sich um die Mittelpunkte der einzelnen Universitätsstandorte. Die leichte Verschiebung zwischen den hier dargestellten Punkten und den tatsächlichen Standorten entsteht durch die teils ungenauen GPS Koordinaten von Google Maps (woraus Rgeoprofile seine Informationen zieht) und Open Street Map (worauf die bisherigen Analysen beruhen). Die Kreise zeigen jeweils potentielle Ursprünge für die Incidents.

Abbildung 3.9a zeigt vier potentielle Standorte für ein Schreibzentrum an. Von Nord-West nach Süd-Ost liegen die Zentren in Garbsen, wo der neue Campus der Fakultät für Maschinenbau gebaut wird, im Bereich des Hauptgebäudes, am ehemaligen Campus am Bismarckbahnhof und in der Nähe von Ruthe, wo sich Anlagen der Tierärztlichen Hochschule und der Gravitationsphysik befinden.

Abbildung 3.9b zeigt eine Detailansicht von Abbildung 3.9a. Der besondere Fokus liegt hierbei auf dem Gebiet rund um das Hauptgebäude, da sich hier die meisten Studierenden aufhalten (siehe Kapitel 3 und Abbildung 3.6).

[5]Für eine detaillierte Beschreibung siehe (Verity et al. 2014, Hauge, Stevenson, Rossmo & Le 2016).



(a) Überblick (b) Detailansicht

Abbildung 3.9: Model Ouput der DPM für die Universitätsstandorte

Im nächsten Schritt sollen die Ergebnisse verfeinert werden. Nun wird nicht ein Standort als ein Incident gezählt, sondern die gerundeten Mittelwerte der CP pro Campus als entsprechende Anzahl an Incidents gewertet. Für Campus 31 ergeben sich so 26 CP aus schriftlichen Leistungen, die dann 26 mal als Incident in das DPM eingehen.

Abbildung 3.10 veranschaulicht die Ergebnisse dieser Analyse. Aufgrund der nunmehr 355 Incidents, im Gegensatz zu den 18 aus dem ersten Durchlauf, ist das Ergebnis deutlich feiner. Da aus verschiedenen Gründen die Standorte in Garbsen, Ruthe und in der Nähe des Bismarckbahnhofs nicht in Frage kommen[6], wird sich nun auf die Standorte in der Nähe des Hauptgebäudes konzentriert.

Der Auswahlbereich ist im Vergleich zu Abbildung 3.9 deutlich kleiner und schärfer umrissen. Wegen der oben beschriebenen Verschiebung werden alle Standorte ein Stück zu weit südlich angezeigt. Wird diese Verschiebung berücksichtigt, liegt das Zentrum der Incidents ein Stück süd-östlich der Kreuzung Schneiderberg/Callinstraße. Da das Schreibzentrum tatsächlich auf der anderen Straßenseite liegt, kann sein Standort auf Basis dieser Analyse als momentan optimal angesehen werden.

[6]Als das Schreibzentrum der LUH gegründet wurde und dementsprechend ein Standort ausgewählt werden musste, befand sich der Campus in Garbsen noch im Bau. Der Standort an der Bismarckstraße war kurz zuvor aufgegeben worden. In Ruthe befinden sich vor allem größere Anlagen und weniger Vorlesungssäle und Seminarräume. Daher erscheint es wenig sinnvoll, an einem dieser Standorte ein Schreibzentrum einer zentralen Einrichtung zu gründen.

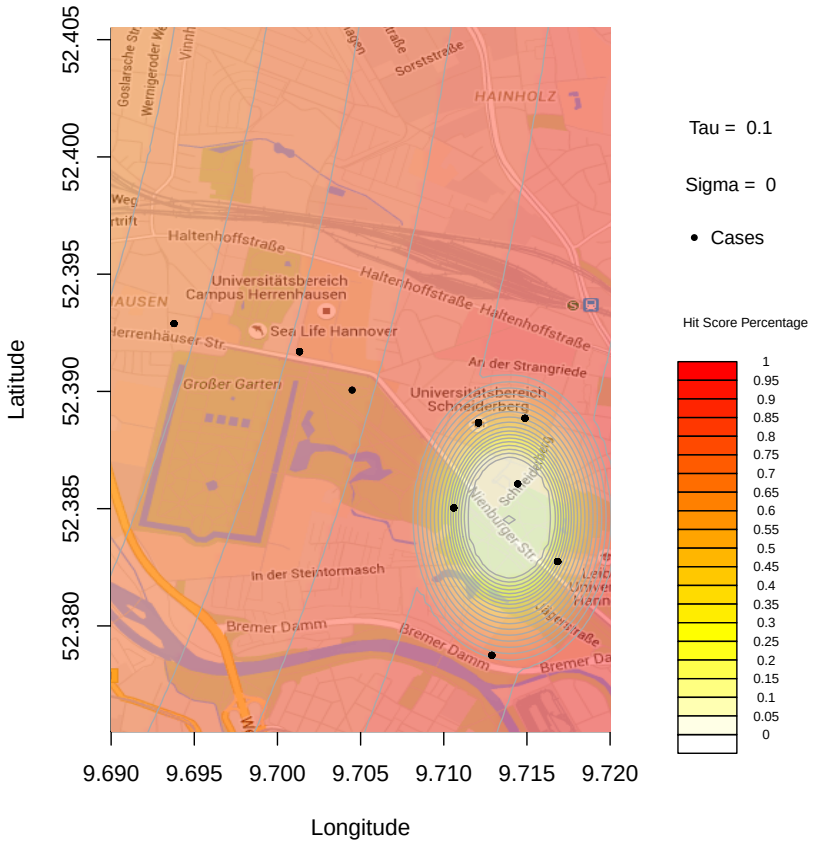


Abbildung 3.10: Model Ouput der DPM für die CP aus schriftlichen Leistungen pro Universitätsstandort

3.2 Beispielhafte quantitative Auswertung von Schreibberatungen

In den letzten Kapiteln wurde auf Basis der Daten der LUH analysiert, wie sich die erwartete Zielgruppe an Studierenden des Schreibzentrums in Bezug auf ihre Fakultätszugehörigkeit und ihre Herkunftsländer darstellt. Da zum Zeitpunkt dieser Arbeit das Schreibzentrum bereits besteht, soll auf Basis der Protokolldaten eines Schreibberaters die Genauigkeit dieser Prognosen abgeschätzt werden.

Der Beobachtungszeitraum wird aus mehreren Gründen auf April 2015 bis April 2016 festgelegt. Das Schreibzentrum befand sich im Jahr 2015 noch in der Gründungsphase, sodass alleinig der Bedarf an Schreibberatung und die geografische Nähe für Studierende ausschlaggebend sein konnten, das Schreibzentrum aufzusuchen. Außerdem handelt es sich bei dem angegebenen Zeitraum um den Hauptarbeitszeitraum des Autors an besagtem Schreibzentrum. Um die Anonymität der Studierenden zu wahren, werden nur aggregierte Daten verwendet. Außerdem ist zu bedenken, dass die Arbeitskapazität des Schreibberaters durch eine 50%-Anstellung entsprechend verringert war.

Insgesamt wurden im besagten Zeitraum 198 Schreibberatungen mit 36 Studierenden durchgeführt. 11 Studierende besuchten dabei nur eine Schreibberatung. Bei diesen Studierenden ist davon auszugehen, dass ihre Erwartungen an eine Schreibberatung nicht erfüllt wurden. Weitere 13 verließen die Schreibberatung nach 5 oder weniger Beratungen. Da in den ersten Sitzungen vor allem auf die Grobstruktur der Arbeit eingegangen wird, kamen keine Korpora zum Einsatz. Bei der Beratung von 10 Studierenden, welche zu mehr als 10 Sitzungen erschienen, wurden auch Rechtschreibung und Grammatik betrachtet. Zum Zeitpunkt der Schreibberatungen befand sich HanConc noch in einer frühen Entwicklungsphase und kam daher nicht zum Einsatz, obwohl es sinnvoll gewesen wäre. Die Erfahrungen und Anforderungen aus diesen Schreibberatungen sind jedoch in das Design von HanConc eingeflossen. Tabelle 3.3 zeigt die deskriptiven Statistiken der Schreibberatungen eines Schreibberaters in Teilzeit über etwa ein Jahr.

Tabelle 3.3: Deskriptive Statistiken der Anzahl an Schreibberatungen pro Studierende

Anzahl Stud.	Anzahl Sbr.	μ	σ	Min	25%	75%	Max
36	198	5,500	5,858	1	1	7,2	23

Tabelle 3.4 zeigt die Schreibberatungen aufgeschlüsselt nach Fakultät und Herkunftsland. Je Fakultät wird auch der Anteil je Herkunftsland angegeben. 146 von 198 Schreibberatungen setzen sich aus der kleinen Anzahl von 5 Kombinationen zusammen. Bei der Kombination Polen und FElt handelt es sich um eine einzige Studierende, die massive Probleme mit der grundsätzlichen Wissenschaftlichkeit ihrer Arbeit hatte und daher sehr viele Sitzungen zu Struktur und Aufbau benötigte. Die hohe Anzahl an FNat und Deutschland ergibt sich aus einem Kooperationsprogramm für Doktorand_innen der Graduiertenakademie der naturwissenschaftlichen Fakultät der LUH mit dem Fachsprachenzentrum, an welches das Schreibzentrum

angeschlossen ist. Die Häufung von Studierenden der Fakultät für Maschinenbau aus China, Tunesien, Vietnam, Iran und Syrien entspricht den Erwartungen aus Kapitel 3.1.1. Auf Grund der dünnen Datenlage wird auf statistische Tests verzichtet.

Tabelle 3.4: Anzahl an Schreibberatungen nach Fakultät und Herkunftsland

	FElt	FElt (%)	FMas	FMas (%)	FNat	FNat (%)	FPhi	FPhi (%)
Brasilien	0	0	9	6,25	4	14,29	0	0
China	0	0	42	29,17	0	0	8	100
Deutschland	0	0	0	0	13	46,43	0	0
Ghana	1	5,56	0	0	0	0	0	0
Iran	0	0	8	5,56	0	0	0	0
Kolumbien	0	0	0	0	6	21,43	0	0
Polen	17	94,44	0	0	5	17,86	0	0
Saudi-Arabien	0	0	1	0,69	0	0	0	0
Syrien	0	0	47	32,64	0	0	0	0
Tunesien	0	0	8	5,56	0	0	0	0
Türkei	0	0	2	1,39	0	0	0	0
Vietnam	0	0	27	18,75	0	0	0	0

3.3 Umfrage unter Schreibberater_innen zum Einsatz von Korpussoftware in Schreibberatungen

Die Zielgruppe von HanConc sind explizit nicht Studierende sondern Schreibberater_innen. Die Software wurde entwickelt, um es ihnen leichter zu machen, Korpora in ihren Schreibberatungen zu nutzen. Daher richtet sich die im Folgenden beschriebene Umfrage an sie. Eine Überprüfung mit Studierenden ist nicht geplant, da hier zu viele Variablen zu berücksichtigen wären[7], um noch den Rahmen dieser Arbeit zu wahren.

Mit der Umfrage werden mehrere Ziele verfolgt: Es soll geklärt werden, auf welcher Ebene Schreibberatungen durchgeführt werden. Wie in Kapitel 2.2 beschrieben, können Schreibberatungen auf unterschiedlichen Abstraktionsebenen durchgeführt werden. Wird nur auf den Prozess des Schreibens und die Grobstruktur des Textes eingegangen und Syntax, Semantik und Rechtschreibung ignoriert, so kann HanConc in seiner jetzigen Form wenig beitragen.

[7]Diese Variablen etwa müssten abgefragt werden:

- Welches pädagogische Konzept hat die Schreibberater_in?
- Welche fachlichen und technischen Voraussetzungen haben die Studierenden?
- Wie gut ist der Zugang zu relevanten Texten?
- Erlauben die Gegebenheiten eine langfristige und tiefergehende Betreuung, wie sie für eine Beratung mit Korpora wahrscheinlich notwendig wäre?
- Wie reagieren Studierende auf alternative Software?
- Wie groß ist der Effekt einzelner Funktionen?

Zusätzlich soll geklärt werden, ob bei den Schreibberater_innen disziplinspezifisches Fachvokabular und grundlegendes Fachverständnis zu den Disziplinen der von ihnen zu beratenden Studierenden vorhanden ist. Erst durch fehlendes Fachwissen bei den Schreibberater_innen und durch die Abwesenheit von geeigneten Nachschlagewerken ergibt sich die Notwendigkeit für eine Nutzung von spezifischen Korpora und dazugehöriger Software.

Außerdem soll erfragt werden, wie die Schreibberater_innen zu linguistischer Komplexität stehen. Hierzu werden einzelne Facetten von HanConc vorgestellt und erfragt, ob diese Funktionen bekannt sind. Des Weiteren wird eruiert, ob Schreibberater_innen in der Lage wären, HanConc selbstständig anzupassen oder dies erlernen zu wollen.

Abschließend wird überprüft, ob eine weitere Verwendung und ggf. Weiterentwicklung von HanConc für die Schreibberater_innen von Interesse wäre.

3.3.1 Methodik

Zielgruppe der Umfrage sind die Schreibzentren aller deutschen Universitäten und Fachhochschulen. Auf Basis einer Liste aller öffentlichen und privaten deutschen Universitäten und Fachhochschulen[8] wurde mittels einer Google-Suche[9] überprüft, ob eine Art von Schreibzentrum oder Schreibberatung angeboten wird. Diese Methode wurde gewählt, da auf diese Weise auch Studierende nach einer Schreibberatung suchen würden.

Für den Fall, dass eine Schreibberatung angeboten wird, wurde die allgemeine E-Mail Adresse des Schreibzentrums oder aber die Adresse der Leiter_in verwendet. Die Schreibzentren wurden dann einzeln angeschrieben und darum gebeten, den Fragebogen an ihre Schreibberater_innen und Tutor_innen weiterzugeben. Um den Datenschutz zu wahren und nicht Prozesse und Mechanismen nach dem Bundesdatenschutzgesetz[10] aufsetzen zu müssen, wurde auf eine Bestätigung bzw. auf einen Rückkanal verzichtet.

Statistische Artefakte werden bei der Umfrage in Kauf genommen. Wenn an Universität A beispielsweise 20 Schreibberater_innen mit Korpora arbeiten und an Fachhochschule B mit lediglich zwei Schreibberater_innen jedoch nicht damit gearbeitet wird, ist das Verhältnis auf Ebene der Schreibberatungen zehn zu eins und auf Einrichtungsebene eins zu eins. Einer Studierenden nützt es jedoch wenig, wenn, statistisch gesehen, Schreibberatungen unter Einsatz von Korpora an der von ihr besuchten Einrichtung durchgeführt werden, dies faktisch jedoch nicht der Fall ist. Wenn der Name der Einrichtung bei der Umfrage ebenfalls erfasst werden würde, wäre vor allem bei kleineren Einrichtungen eine Personenbeziehbarkeit gegeben, sodass das Bundesdatenschutzgesetz wieder Anwendung finden würde. Um hier unnötigen Aufwand

[8] Die Liste wurde folgender Seite entnommen:
https://de.wikipedia.org/wiki/Liste_der_Hochschulen_in_Deutschland (Stand: 30. Mai 2021)

[9] Es wurde nach Universität oder Fachhochschule, dem jeweiligen Ort, Schreibberatung und Schreibzentrum gesucht. Jeweils die ersten vier Ergebnisseiten wurden berücksichtigt.

[10] Das Bundesdatenschutzgesetz bezieht sich in §1 auf die Richtlinie (EU) 2016/680. Diese Richtlinie findet Anwendung, wenn Daten „personenbezogen" sind (Artikel 1). Personenbezogen sind Daten, wenn mit ihrer Hilfe natürliche Personen identifiziert werden können (Artikel 3). Um für personenbezogene Daten notwendige „technische und organisatorische Maßnahmen" zu verhindern (Artikel 4 (1)f), werden Fragen so formuliert, dass ein Personenbezug nicht möglich ist.

zu vermeiden, wird daher eine Ungenauigkeit auf Einrichtungsebene in Kauf genommen.

Der eigentliche Fragebogen wurde mit Google Forms erstellt. Auf diese Weise ergibt sich der Vorteil, dass der Fragebogen kostenlos und einfach anpassbar ist. Außerdem werden die Ergebnisse in einem auswertbaren Format gespeichert. Über Logiken innerhalb des Fragebogens wurden Absprungpunkte definiert. So ist etwa die Frage, ob die Schreibberater_in programmieren lernen würde, um HanConc an ihre Anforderungen anzupassen, unnötig, wenn ganz am Anfang schon klargestellt wurde, dass kein Interesse an korpuslinguistischer Unterstützung besteht. Die tatsächlichen Fragen befinden sich im Anhang.

Inhaltlich zielen die Fragen auf die Annahmen aus Kapitel 1 und Kapitel 2 ab. Es soll überprüft werden, ob die Schreibzentren ähnlich aufgestellt sind wie das der LUH. Dies bedeutet, dass wenige Schreibberater_innen mit einem geisteswissenschaftlichen Hintergrund eine Vielzahl von Studierenden unterschiedlichster Fachrichtungen und muttersprachlicher Hintergründe begleiten. In einem zweiten Schritt wird erfragt, ob die Schreibberatungen so tief in die studentischen Texte einsteigen, dass sich der Einsatz von Hilfsmitteln auf Wortebene lohnt. In der nächsten Fragengruppe werden Fragen zum konkreten Vorgehen bei Schreibberatungen gestellt. Ziel ist es hier, zu erfahren, ob der Einsatz von Korpora bereits stattfindet oder angedacht ist. Sollten bereits Korpora verwendet werden, wird ermittelt, wie diese eingesetzt werden. Im letzten Fragenblock wird eruiert, ob und wie sich die Teilnehmenden der Umfrage den Einsatz von HanConc vorstellen können.

3.3.2 Schreibzentren in Deutschland

Grundlage für die Analyse der Schreibzentren in Deutschland ist eine Auflistung aller Universitäten, Fachhochschulen (FH), dualen (DH) und pädagogischen Hochschulen (PH). Berücksichtigt werden Einrichtungen unterschiedlicher staatlicher, privater und kirchlicher Träger. Von den etwa 430 aufgelisteten Einrichtungen wurden nur solche betrachtet, die mehr als 2.000 Studierende haben. Diese Einschränkung reduziert die Anzahl an Einrichtungen auf 214. Es wurden nur Schreibzentren berücksichtigt, die allen Studierenden zur Verfügung stehen, d.h. dass Schreibzentren ausschließlich für einzelne Institute nicht berücksichtigt wurden. Außerdem wurden nur Schreibzentren aufgenommen, die Schreibberatungen, Schreibwerkstätten oder Lange Nächte der aufgeschobenen Hausarbeiten, Schreibcoachings, Peertutoren oder ähnliches anbieten[11]. Schreibzentren, deren Projekte bereits ausgelaufen oder nicht mehr erreichbar sind, wurden ignoriert.

Von den 214 Einrichtungen haben 115 zum Stand Juni 2021 ein Schreibzentrum oder ein ähnliches Angebot. Abbildung 3.11 zeigt die Anzahl an Einrichtungen mit und ohne Schreibzentrum je Bundesland. Auffällig ist hier, dass es anscheinend keinen Unterschied macht, ob die Einrichtungen in einem Flächenbundesland oder einem Stadtstaat liegen. So haben Niedersachsen und Bremen etwa deutlich mehr Bildungseinrichtungen mit Schreibzentrum als ohne. Im Gegensatz dazu ist die Situation in Rheinland-Pfalz und Berlin genau andersherum. Auch

[11] Um eine bessere Lesbarkeit zu gewährleisten, werden alle diese Angebote unter dem Begriff Schreibzentrum zusammengefasst.

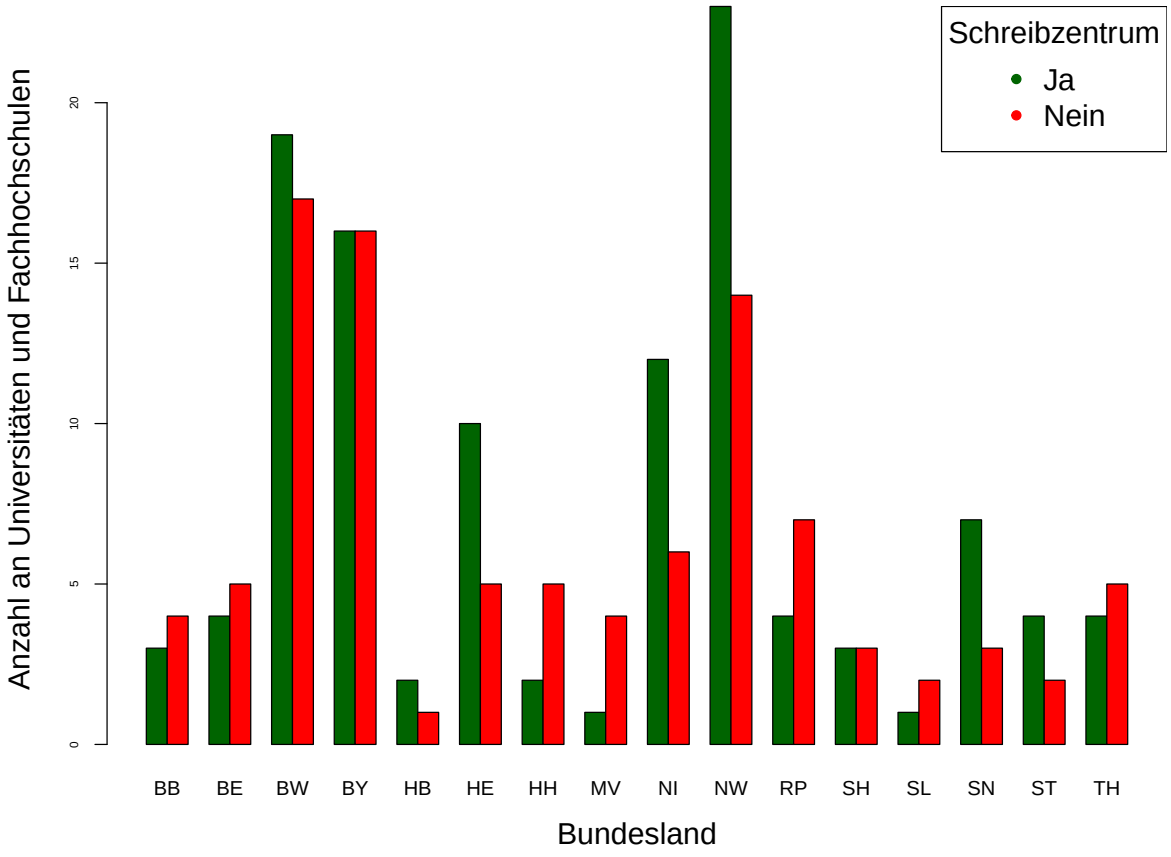


Abbildung 3.11: Anzahl an Einrichtungen mit und ohne Schreibzentrum nach Bundesland

beim Vergleich des Südens gegenüber dem Norden oder den alten gegenüber den neuen Bundesländern können keine signifikanten Unterschiede erkannt werden.

Abbildung 3.12 (links) zeigt die Anzahl an Schreibzentren je Form und Träger[12]. Bei der Form der Einrichtung zeigt sich vor allem, dass Schreibzentren ein Service von Universitäten und pädagogischen Hochschulen sind. Es liegt die unbewiesene Vermutung nahe, dass dies mit dem höheren akademischen Anspruch und dem höheren Schreibaufwand an Universitäten zusammenhängt.

Eindeutig ist jedoch der Trend bei der Anzahl an Schreibzentren nach Träger in Abbildung 3.12 (rechts). An keiner der 19 privaten Fachhochschulen gibt es ein Schreibzentrum; bei den konfessionellen Einrichtungen zumindest an einer (Katholische Universität Eichstätt-Ingolstadt). Bei den staatlichen Fachhochschulen besitzt die Hälfte ein Schreibzentrum, wohingegen 74% der Universitäten ein Schreibzentrum anbieten. Gründe für diese Verteilung zu finden, ist nicht Teil dieser Arbeit.

Abbildung 3.13 zeigt die Analyse der Anzahl an Studierenden. Teil (a) zeigt, dass Universitäten grundsätzlich um ein vielfaches größer sind als andere Einrichtungen. Zu den größeren nicht-universitären Einrichtungen gehören vor allem überregionale private Fachhochschulen.

[12]Zusätzlich ist in dem Datensatz noch die Variable „Promotionsrecht" vorhanden. Auf diese wurde allerdings wegen der hohen Korrelation mit der Form der Einrichtung verzichtet.

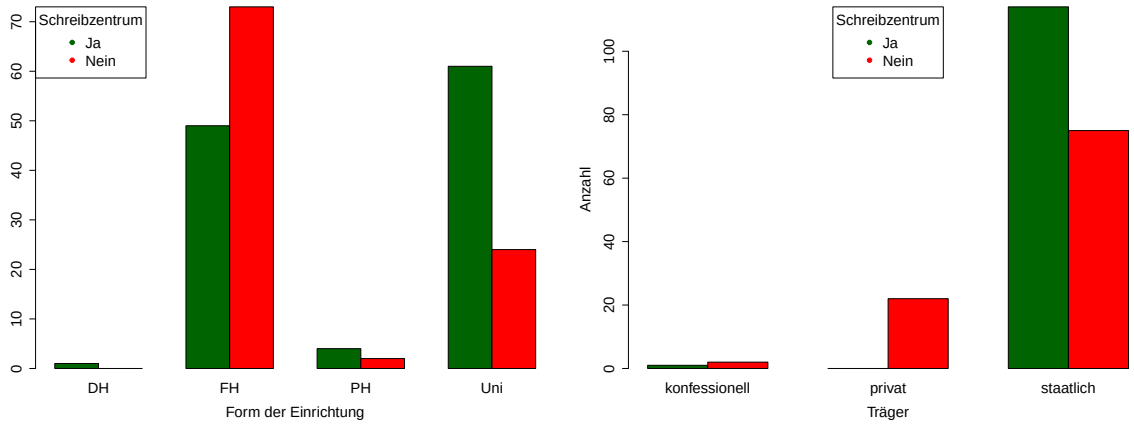
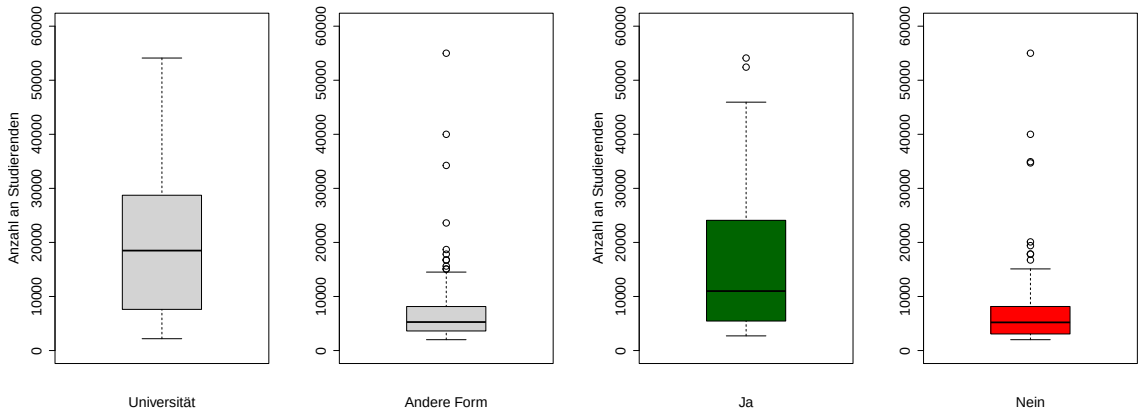


Abbildung 3.12: Anzahl an Einrichtungen mit und ohne Schreibzentrum nach Form und Träger



(a) Anzahl an Studierenden nach Form der Einrichtung

(b) Anzahl Studierende pro Einrichtung nach Vorhandensein eines Schreibzentrums

Abbildung 3.13: Analyse der Anzahl an Studierenden pro Einrichtung nach Vorhandensein eines Schreibzentrums

Teil (b) derselben Abbildung zeigt die Anzahl an Studierenden pro Einrichtung mit und ohne Schreibzentrum. Die Einrichtungen mit Schreibzentrum sind im Durchschnitt in etwa doppelt so groß wie die Einrichtungen ohne Schreibzentrum. Dieser Unterschied wurde mit einem Welch Two Sample t-Test überprüft und ist statistisch signifikant.

Zusammenfassend ist festzuhalten, dass ein Schreibzentrum statistisch gesehen vor allem eine universitäre Einrichtung in staatlicher Trägerschaft ist. Ungeklärt bleibt, ob sich bei den anderen Einrichtungen auf Grund ihrer Größe oder ihrer Form kein Schreibzentrum lohnt oder sie wegen einer praktischeren Ausrichtung darauf verzichten.

3.3.3 Ergebnisse der Umfrage unter Schreibberater_innen

Zum Stichtag 16. September 2021 haben 65 Schreibberater_innen den Fragebogen ausgefüllt. Es muss bedacht werden, dass nicht alle Fragen beantwortet werden mussten. Der Fragebogen ist mit Absicht so konstruiert worden, da somit zumindest für einige Fragen Antworten vorlie-

gen, auch wenn der Fragebogen nicht bis zum Ende ausgefüllt wurde.

Einige Fragen haben eine Mehrfachantwort erlaubt. Bei der Besprechung der Ergebnisse in diesem Text wird darauf eingegangen, ob die befragten Schreibberater_innen eine bestimmte Antwort auch ausgewählt haben. Die Abbildungen hingegen zeigen das Verhältnis von allen ausgewählten Antworten[13].

Der erste Fragenblock beschäftigt sich mit dem organisatorischen Kontext der Schreibzentren. Die Teilnehmenden der Umfrage arbeiten dabei im Verhältnis drei zu zwei eher an einer Universität als an einer Fachhochschule. Etwa ein Viertel der Einrichtungen (23,8%/Abbildung 3.14a) sind an eine Fakultät oder einen Fachbereich angebunden. Der große Rest besteht als Teil einer Serviceeinrichtung wie einer Graduiertenakademie, einem allgemeinen Sprachzentrum oder einer Bibliothek. Es werden vor allem Betreuungen auf Deutsch, Deutsch als Fremdsprache und Englisch angeboten (Abbildung 3.14b). Die Umfrage ergab, dass Schreibberater_innen vor allem als frei oder fest angestellte Mitarbeiter_innen tätig sind. Hinzu kommen noch einige studentische Schreibberarter_innen. Zusätzlich arbeiten an etwa 41% der Schreibzentren noch Koordinator_innen oder Mitarbeiter_innen der Administration. Dezidierte IT Mitarbeiter_innen gibt es nur an drei Schreibzentren (Abbildung 3.14c). 57% der Schreibzentren haben weniger als vier Mitarbeiter_innen, weitere 24% vier bis acht und 19% sind mit mehr als acht Mitarbeiter_innen ausgestattet[14] (Abbildung 3.14d). Für das typische Schreibzentrum gilt somit, dass dort wenige Vollzeitkräfte und einige studentische Hilfskräfte beschäftigt werden, Deutsch, Deutsch als Fremdsprache und Englisch angeboten werden und es eher fakultätsübergreifend angesiedelt ist. Dies deutet darauf hin, dass eine Spezialisierung auf einzelne Fachrichtungen kaum möglich ist und somit Schreibberater_innen auch außerhalb ihrer fachlichen Expertise beraten können müssen.

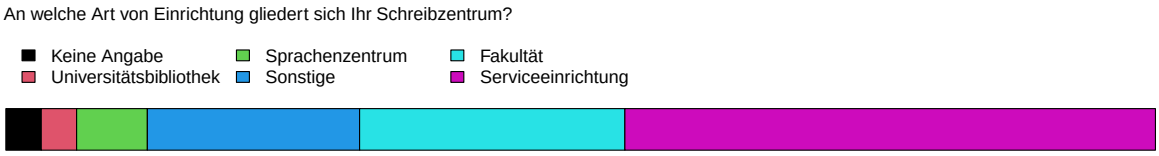
Der zweite Fragenblock widmet sich dem akademischen Werdegang der Schreibberater_innen. Mit 40% (26 Schreibberater_innen) hat der Großteil promoviert, während vier bzw. sechs Schreibberater_innen einen Bachelor oder ein Diplom vorweisen können. Mit je 14 haben die übrigen Schreibberater_innen ihr Studium mit einem Master oder Magister abgeschlossen (Abbildung 3.15a). Bis auf wenige Ausnahmen haben alle übrigen Schreibberater_innen einen geistes- oder sozialwissenschaftlichen Hintergrund (Abbildung 3.15b).

Die Befragten sollten angeben, welchen akademischen Hintergrund ihre Studierenden haben und konnten dabei mehrere Fachrichtungen angeben. In etwa gleichstarken Gruppen sind Geistes- und Sozialwissenschaftler_innen, Ingenieurwissenschaftler_innen, sowie Wirtschafts- und Naturwissenschaftler_innen vertreten. Jede dieser Gruppen wird an zwei Drittel bis drei Viertel der Schreibzentren beraten. Mediziner_innen besuchen etwa 30% der Schreibzentren. Andere Fachrichtungen sind nicht oder nur als Einzelfall vertreten (Abbildung 3.16).

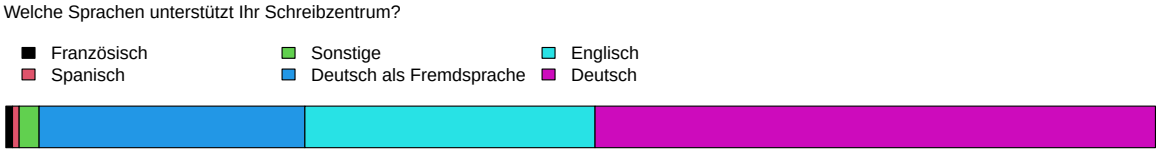
Abbildung 3.17 zeigt die Verteilung der durch die Schreibberater_innen vermuteten Erst-

[13] Eine Schreibberaterin hat etwa bei Frage 5 A, B und D ausgewählt. Im Text zählt sie zu den x%, die auch B ausgewählt haben. In den Abbildungen werden alle As, Bs, Cs und Ds ausgezählt und in Prozentwerte umgerechnet.

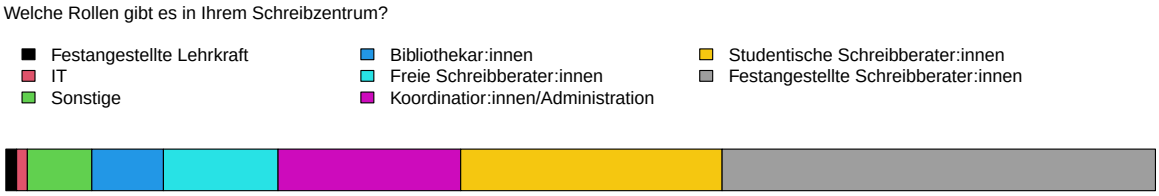
[14] Es sei an dieser Stelle erwähnt, dass diese Zahlen ggf. durch die Anonymität der Ausfüllenden verzerrt werden. Etwa die Frage nach der Zugehörigkeit zu einer Universität oder Fachhochschule kann durch mehrere Ausfüllende an derselben Einrichtung beeinflusst werden.



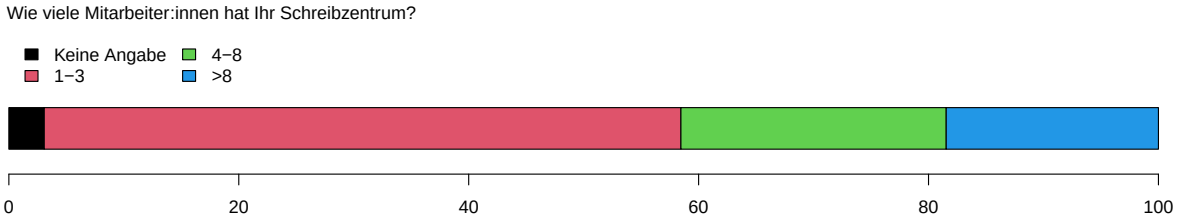
(a) Arbeitsort der Schreibberater_innen (Angaben in %)



(b) Unterstützte Sprachen in den Schreibberatungen (Angaben in %)

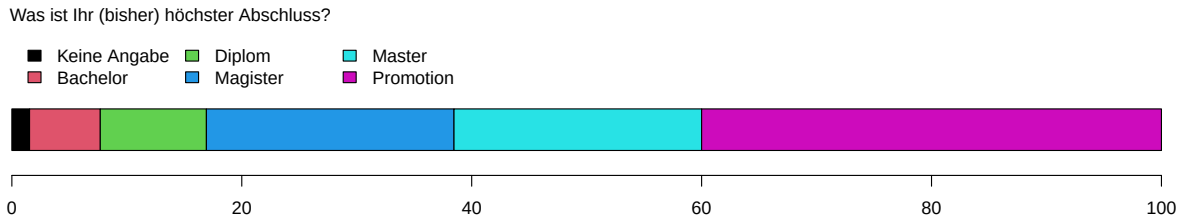


(c) Vorhandene Arbeitsposition im Schreibzentrum (Angaben in %)

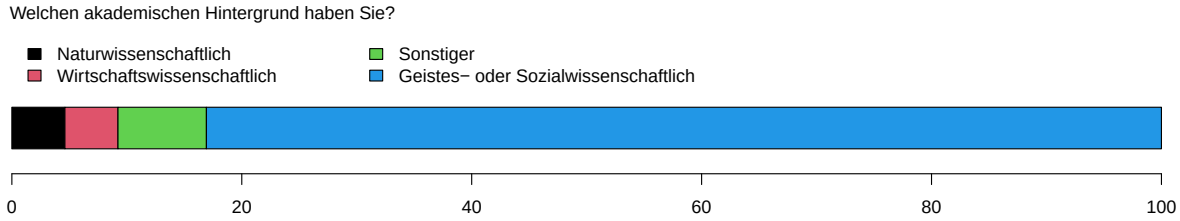


(d) Anzahl an Mitarbeiter_innen (Angaben in %)

Abbildung 3.14: Grundlegende Charakteristika der befragten Schreibzentren



(a) (Bisher) höchster Abschluss der Schreibberater_innen (Angaben in %)



(b) Akademische Fachrichtung der Mitarbeiter_innen (Angaben in %)

Abbildung 3.15: Akademischer Hintergrund der Mitarbieter_innen

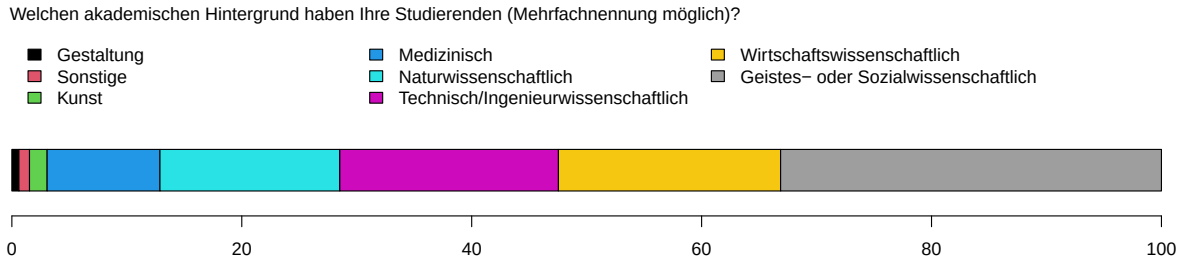


Abbildung 3.16: Akademischer Hintergrund der betreuten Studierenden (Angaben in %)

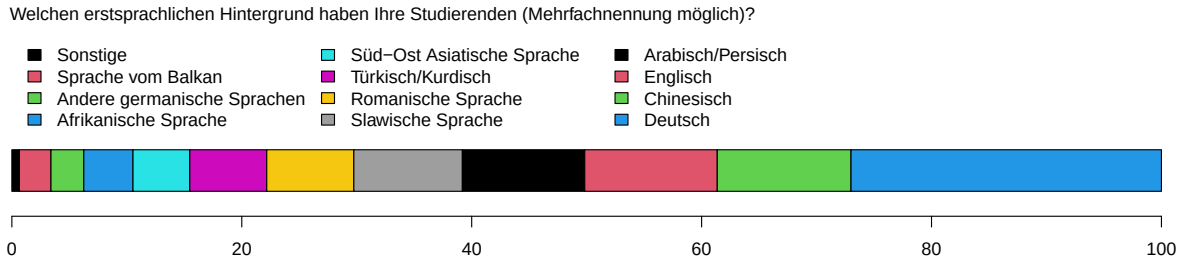


Abbildung 3.17: Vermutete Erstsprache der betreuten Studierenden (Angaben in %)

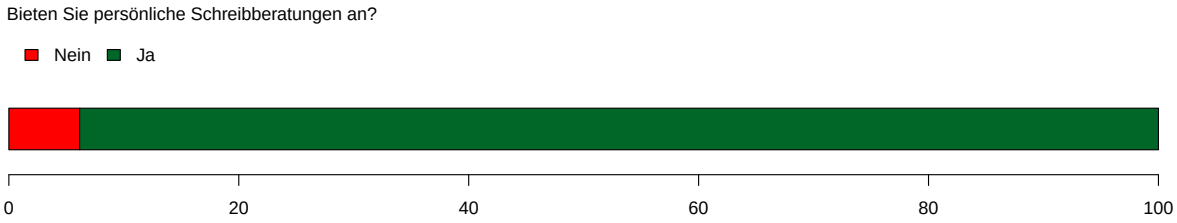
sprache ihrer Studierenden[15]. Die Erstsprachen waren vorgegeben und es gab die Möglichkeit einer Mehrfachauswahl. Es muss an dieser Stelle darauf hingewiesen werden, dass solche biografischen Daten wahrscheinlich nicht systematisch von den Schreibberater_innen erfasst werden und die Anzahl der Nennungen nichts über die quantitative Zusammensetzung aussagt. Es kann also sein, dass Deutsch und Chinesisch gleichberechtigt genannt werden, das Verhältnis jedoch 99 zu eins ist. Die Auswahlliste wurde mit Blick auf den Fragebogen erstellt, um Tendenzen in der Herkunft der Studierenden zu erkennen und weniger, um eine möglichst genaue Kartierung von Erstsprachen zu gewährleisten.

Die Verteilung der Erstsprachen zeigt wenige Überraschungen. Mit Ausnahme von zwei Schreibzentren werden überall Studierende mit Deutsch als Erstsprache betreut. Die beiden Abweichungen sind wahrscheinlich auf einzelne Schreibberater_innen zurückzuführen, die dezidiert englischsprachige Beratungen anbieten[16]. Bei den übrigen Erstsprachen zeigen sich vier Trends. Die Sprachen der deutschen Anrainerstaaten (also Polnisch oder Französisch) sind häufig vertreten. Ebenso häufig werden Sprachen der größten Einwanderungsgruppen (also Arabisch und Türkisch) genannt. Chinesische Studierende besuchen ebenfalls oft Schreibzentren. Sprachen aus kleineren Ländern, etwa vom Balkan oder aus Ländern, die tendenziell wenige Studierende nach Deutschland schicken (etwa aus Afrika), sind entsprechend selten vertreten. Die Ergebnisse decken sich mit der Annahme aus Kapitel 1.3, dass eine sprachliche Barriere zwischen Schreibberater_innen und Studierenden existiert.

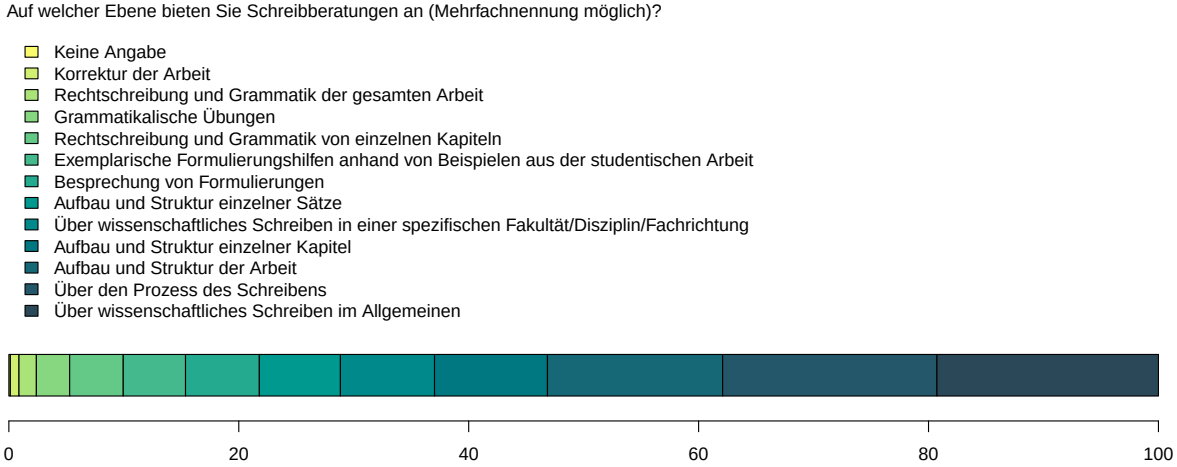
Von 65 Schreibberater_innen bieten 61 individuelle Schreibberatungen an (Abbildung 3.18a) und nur diese haben die Fragen dazu auch beantwortet. Die Antwortmöglichkeiten zur Frage, welche sich mit der Ebene, auf der Schreibberatungen angeboten werden, beschäftigt, steigern sich im Detailgrad von Beratungen über das wissenschaftliche Schreiben im Allgemeinen bis zu einer kompletten Korrektur der Arbeit (Abbildung 3.18b). Mit steigendem Detailgrad der Beratungen sinkt die Anzahl an Schreibberater_innen, die solche Leistungen anbieten. Immerhin noch etwa 60% der Schreibberater_innen bieten Hilfen zu Formulierungen ausgewählter Kapi-

[15]Die Kategorie „Andere germanische Sprachen" beinhaltet im Fragebogen Dänisch und Niederländisch; unter „Romanische Sprachen" werden Französisch und Spanisch aufgelistet; zu „Slawischen Sprachen" zählen im Fragebogen Polnisch und Russisch; bei „Afrikanischen Sprachen" wird darauf hingewiesen, dass die Sprachen der ehemaligen Kolonialmächte nicht gemeint sind; „Süd-Ost asiatische Sprachen" umfassen etwa Vietnamesisch oder Koreanisch. Die Ungenauigkeit wird hier akzeptiert, um den Fragebogen kompakt zu halten.

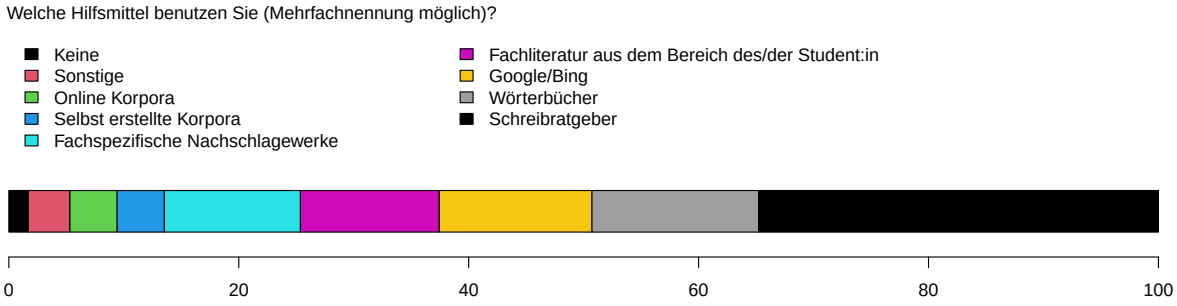
[16]Das Schreibzentrum der LUH bietet etwa Schreibberatungen durch Erstsprecher_innen des Englischen an.



(a) Vorhandensein von persönlichen Schreibberatungen (Angaben in %)



(b) Ebene der Schreibberatungen (Angaben in %)



(c) Verwendete Hilfsmittel (Angaben in %)

Abbildung 3.18: Grundlegende Charakteristika der angebotenen Schreibberatungen

tel an. Knapp 20% ermöglichen dies auch für die gesamte Arbeit. Für diese Beratungen werden von 88% der Schreibberater_innen Schreibratgeber verwendet und etwa die Hälfte nutzt Wörterbücher und Nachschlagewerke. Am interessantesten für potentielle Nutzer_innen von HanConc sind die 25%, die online oder eigene Korpora für ihre Beratungen nutzen (Abbildung 3.18c). Es ergibt sich für HanConc also eine definitive Zielgruppe von 25% und eine mögliche Zielgruppe von 50% aller Schreibberater_innen. Für erstere könnten sich komplexere und damit wahrscheinlich genauere Funktionen eignen, während letztere vor allem über die einfache Integration von für sie relevanten Texten angesprochen werden könnten.

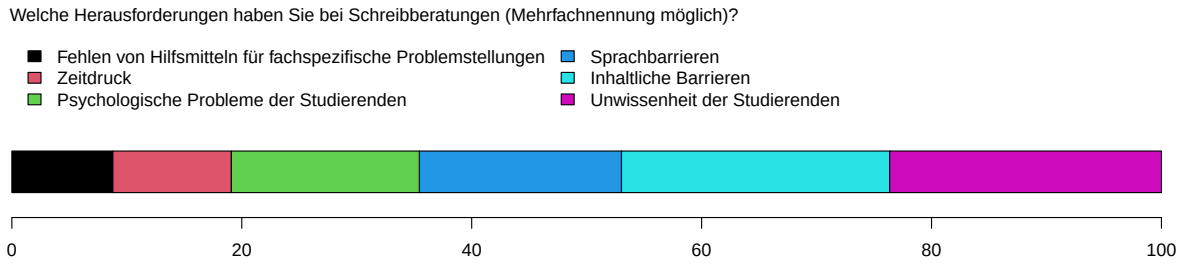
Als relevante Herausforderungen[17] für ihre Schreibberatung wählten die Schreibberater_innen in absteigender Reihenfolge inhaltliche Barrieren (62%), Sprachbarrieren (49%) bzw. fehlende Hilfsmittel (29%) aus (Abbildung 3.19a). Auch wenn diese Antworten suggerieren, dass vor allem fachliche und inhaltliche Barrieren eine erfolgreiche Schreibberatung behindern, so scheinen doch vor allem technische Fähigkeiten im Zusammenspiel mit fehlenden Texten einem erfolgreichen Einsatz von Korpora entgegenzustehen (Abbildung 3.19b). Die Umfrage hat zusätzlich ergeben, dass fast ausschließlich online verfügbare Korpora zur Schreibberatung genutzt werden[18]. Es findet sich kein Hinweis auf selbst erstellte Textsammlungen bzw. Korpussoftware in den Umfrageergebnissen. Ein grundsätzliches korpuslinguistisches Verständnis scheint jedoch vorhanden zu sein, da eine Mehrheit Key Words in Context (KWIC), Kollokationen und Häufigkeitsverteilungen kennt. Selbst informationstechnologische Ergebnistypen wie n-Grams, Latent Semantic Analysis (LSA) und Word Embeddings sind einigen Schreibberater_innen bekannt (Abbildung 3.19c).

Mit dem letzten Fragenblock wurde überprüft, ob Schreibberater_innen bereit sind, sich mit HanConc als Software auseinanderzusetzen. Obwohl 83% der Schreibberater_innen die Software anpassen möchten, würden 83% zwar einen Nachmittag aber nur 25% drei Tage investieren, um zu lernen, wie HanConc an die jeweiligen Bedürfnisse angepasst werden kann. Aus den letzten drei Fragen ergibt sich, dass HanConc zwar gerne genutzt werden würde (Abbildung 3.20a), allerdings nur, wenn die Einstiegshürden durch entlastende Dokumentation, Schulungen und ein online Deployment möglichst gering gehalten werden. Unter diesen Bedingungen könnten sich über 60% vorstellen, HanConc langfristig in ihren Schreibberatungen einzusetzen (Abbildungen 3.20b, 3.20c und 3.20d).

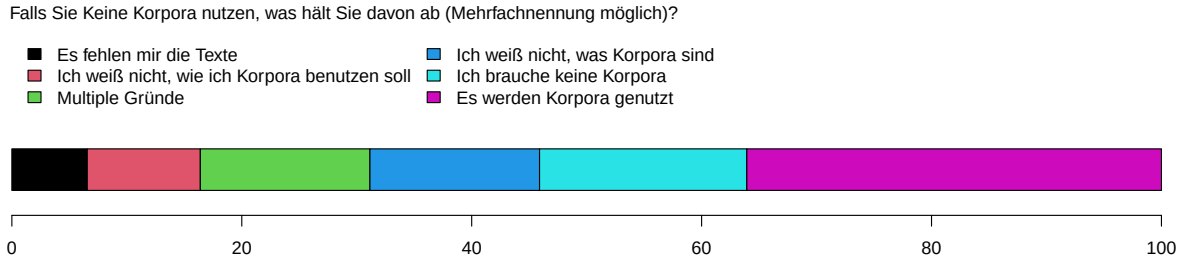
Insgesamt hat die Umfrage unter deutschen Schreibberater_innen gezeigt, dass die Annahmen aus Kapitel 1.3 zutreffend sind. Es gibt eine Zielgruppe für HanConc, welche mit einer einsteigerfreundlichen aber gleichzeitig wirksamen Software für den Einsatz von Korpora in Schreibberatungen unterstützt werden kann (Abbildung 3.21). Fachliche, sprachliche und inhaltliche Barrieren, die sich zwischen Schreibberater_innen und Studierenden ergeben, könnten dadurch abgebaut und somit die Effektivität von Schreibberatung erhöht werden.

[17] Die übrigen Antworten sind für diese Arbeit irrelevant, da etwa psychologischer Druck nicht durch korpuslinguistische Software beeinflusst werden kann.

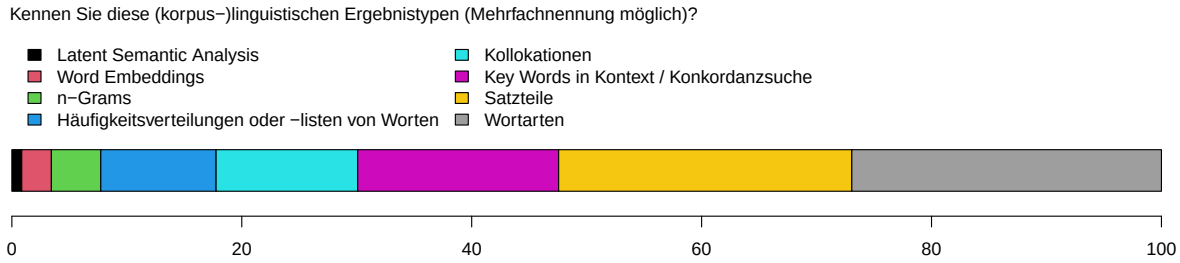
[18] Etwa Corpus of Contemporary American English (COCA) oder Digitales Wörterbuch der deutschen Sprache (DWDS); die komplette Liste befindet sich im Anhang.



(a) Herausforderungen bei Schreibberatungen (Angaben in %)

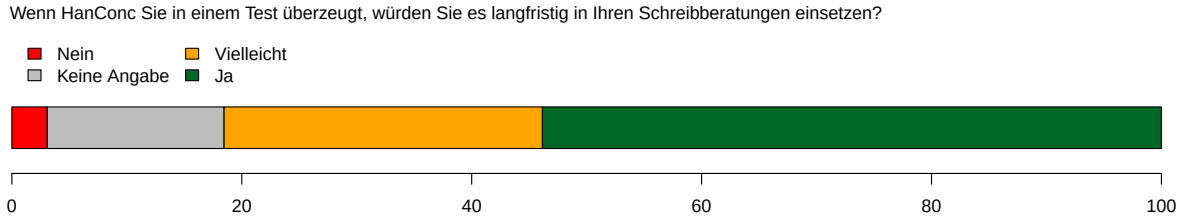


(b) Bisherige Gründe, die für oder gegen den Einsatz von Korpora sprechen (Angaben in %)

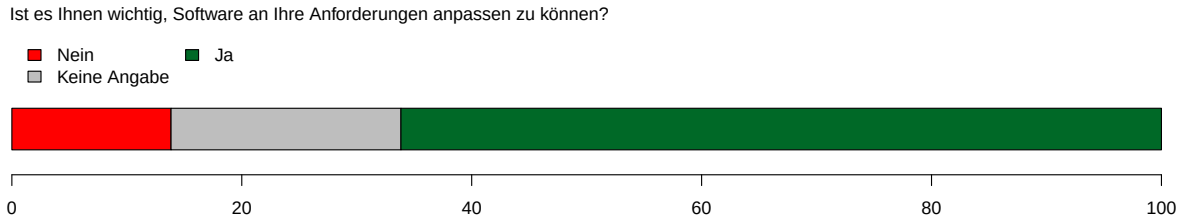


(c) Kenntnisse über linguistische Verfahren (Angaben in %)

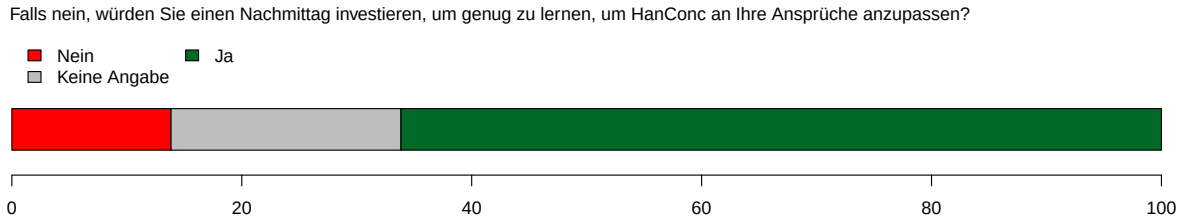
Abbildung 3.19: Anforderungen an die Nutzung von Korpora



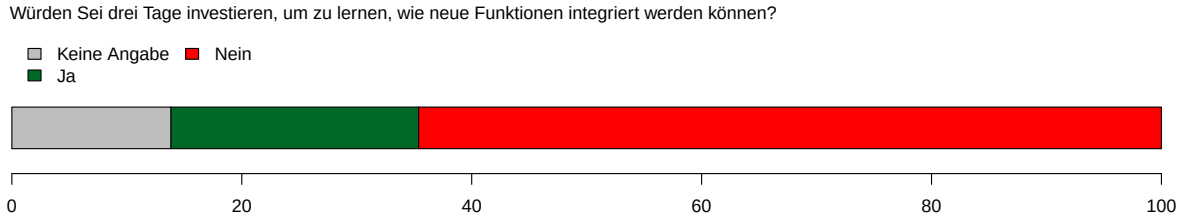
(a) Grundsätzliche Meinung zum Einsatz von HanConc im eigenen Schreibzentrum (Angaben in %)



(b) Bewertung der Möglichkeit Korpussoftware anpassen zu können (Angaben in %)



(c) Würde ein Nachmittag investiert werden, um zu lernen, wie HanConc angepasst werden kann (Angaben in %)



(d) Würden drei Tage investiert werden, um zu lernen, wie HanConc angepasst werden kann (Angaben in %)

Abbildung 3.20: Bereitschaft die notwendigen Kenntnisse zur eigenständigen Weiterentwicklung von HanConc zu erlernen

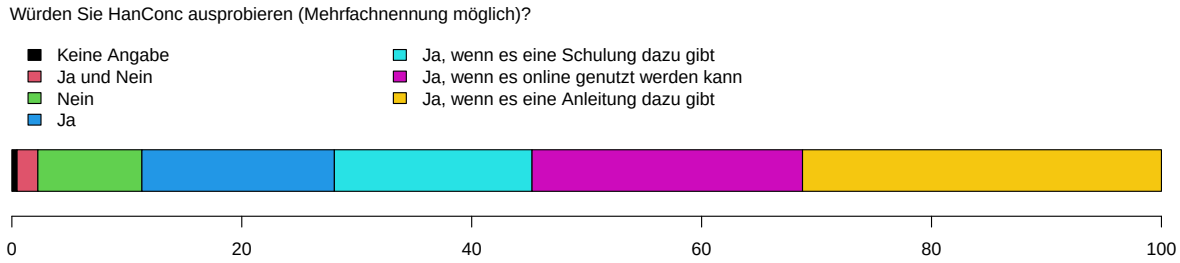


Abbildung 3.21: Bereitschaft HanConc auszuprobieren (Angaben in %)

3.4 Zwischenfazit

Kapitel 3.1 hat die Zielgruppe für Schreibberatungen an der LUH beschrieben. Zur Zeit der Gründung des Schreibzentrums an seinem tatsächlichen Standort sind vor allem Studierende aus Iran, Syrien, Tunesien, China und Vietnam als Teilnehmer_innen von Schreibberatungen zu erwarten gewesen. Auf Grund der geografischen Nähe zu vielen Gebäuden der Fakultäten für Maschinenbau, Bauingenieurwesen und Elektrotechnik & Informatik muss vermehrt mit Studierenden aus diesen Fakultäten als Nutzer_innen von Schreibberatung gerechnet werden. Eine Besonderheit ergibt sich für die naturwissenschaftliche Fakultät, da deren Graduiertenakademie durch Vereinbarungen mit dem Fachsprachenzentrum vorrangig Kapazitäten beanspruchen kann. Basierend auf diesen Analysen muss davon ausgegangen werden, dass Schreibberater_innen mit unterschiedlichsten Erstsprachen und akademischen Traditionen konfrontiert werden und damit zusätzliche Unterstützung im Umgang mit den sprachlichen und inhaltlichen Herausforderungen ihrer Schreibberatungen benötigen. Korpora und Korpussoftware könnten Mittel sein, diesen Herausforderungen zu begegnen.

Kapitel 3.2 hat die tatsächlichen Erfahrungen eines Schreibberaters am Schreibzentrum der LUH ausgewertet und herausgearbeitet, dass diese mit den zuvor getroffenen Annahmen übereinstimmen. In Bezug auf das Schreibzentrum der LUH erscheint eine Fokussierung auf Studierende der Ingenieurwissenschaften auf Grund der Ergebnisse daher sinnvoll.

Eine Umfrage unter Schreibberater_innen hat laut Kapitel 3.3 ergeben, dass sich die Annahmen aus Kapitel 1.3 und die antizipierten Herausforderungen im Umgang mit Studierenden aus unterschiedlichsten Fachrichtungen und Herkunftsländern ebenfalls mit den Erfahrungen an anderen Schreibzentren decken. Außerdem ist eine grundsätzliche Bereitschaft, sich mit Korpora, Korpussoftware und damit auch HanConc zu beschäftigen, zu erkennen.

Die Ergebnisse dieses Kapitels dienen als Grundlage für die folgenden Überlegungen. Für die unterschiedlichen Fakultäten sollen Korpora angelegt sowie bestehende Korpussoftware analysiert und eine Software entsprechend der hier formulierten Anforderungen programmiert werden.

Kapitel 4

Aufbau und Struktur des Hannover Advanced Academic Writing Corpus (HAAWC)

Wie bereits in der Einleitung und den vorherigen Kapiteln beschrieben, sollen in dieser Arbeit Schreibberatungen als zu optimierendes Forschungsobjekt dienen. Die letzten Kapitel haben aufgezeigt, warum es sich lohnt Schreibberatungen anzubieten und wie Studierende von ihnen profitieren können. Gleichzeitig wurden die Herausforderungen aufgezeigt, die mit Hilfe vom Hannover Concordancer (HanConc) als technische Lösung überwunden werden sollen. Damit HanConc als Korpussoftware wirksam werden kann, müssen Texte als Grundlage gesammelt, aufbereitet und integriert werden. Dieses Kapitel beschreibt die zusammengestellte Textsammlung im Allgemeinen und geht wegen der Erkenntnisse aus Kapitel 3 im Besonderen auf die ingenieurwissenschaftlichen Sub-Korpora ein. Die jeweiligen Analysen werden mit statistischen und korpuslinguistischen Kennzahlen untermauert.

4.1 Dissertationen als textuelle Grundlage

Schreibberatungen sollen Studierenden helfen, ihre akademischen Abschlussarbeiten zu schreiben. Daher muss die Grundlage für die Korpora, die ihnen dabei helfen sollen, aus einem universitären oder Forschungskontext stammen. Für Arbeiten, die vor einer Dissertation geschrieben werden, d.h. Bachelor-, Diplom- und Masterarbeiten, ergibt sich das Problem, dass diese nicht systematisch qualitätsgesichert gesammelt und archiviert werden. Forschungspaper aus entsprechenden Zeitschriften sind größtenteils zugangsbeschränkt oder dürfen für die hier intendierten Zwecke nicht verwendet werden. Freie Texte wie etwa von *arXiv* unterliegen keiner ausreichenden Qualitätskontrolle, sodass keine ausreichend hohe Qualität gewährleistet werden kann. Stattdessen werden für HanConc Promotionen verwendet, da diese im Verlauf ihrer Veröffentlichung von mindestens zwei Professor_innen überprüft werden. Außerdem haben diese Texte den Vorteil, dass sie vom Reifegrad der Schreibenden näher bei den Studierenden und den von ihnen zu schreibenden Arbeiten liegen als Veröffentlichungen von Forscher_innen mit

teils jahrzehntelanger Erfahrung.

Für die Leibniz Universität Hannover (LUH) übernimmt die Technische Informationsbibliothek und Universitätsbibliothek (TIB/UB) die Speicherung, Organisation und Veröffentlichung von Dissertationen. Daher wurde deren Dissertationskatalog zum Stand Oktober 2016 als textliche Grundlage für diese Arbeit verwendet.

Da nicht abschließend geklärt werden kann, ob Textmining für wissenschaftliche und pädagogische Zwecke, wie in dieser Arbeit beschrieben, legal ist, wird auf eine Veröffentlichung der Liste verzichtet. Jedoch kann die Liste beim Autor erfragt werden (siehe auch Kapitel 5.1).

4.2 Deskriptive Beschreibung des Gesamtkorpus

Zunächst ist zwischen dem Hannover Advanced Academic Writing Corpus (HAAWC) und dem Hannover Concordancer (HanConc) zu unterscheiden. Bei ersterem handelt es sich um eine Textsammlung bestehend aus Dissertationen der LUH, während letzteres ein dort entwickeltes Programm zur Analyse eben selbiger Texte ist. HAAWC kann auch mit anderer Software untersucht werden und HanConc mit anderen Korpora genutzt werden.

Tabelle 4.1 zeigt eine Zusammenfassung von HAAWC. Sowohl die Anzahl der Dissertationen pro Fakultät als auch der Seitenumfang ergeben sich aus der von der TIB/UB zur Verfügung gestellten Datenbank. Die übrigen Variablen wurden nach dem Part-of-Speech Tagging erhoben.

Tabelle 4.1: Deskriptive Zusammenfassung von HAAWC

	Anzahl	Wortanzahl	Wortanzahl (μ)	Umfang in Seiten (μ)	Wörter pro Seite (μ)
FArc	26	1.972.736	75.874,46	230,84	316,64
FBau	98	4.112.603	41.965,34	182,91	231,98
FElt	50	2.152.712	43.054,24	176,50	245,21
Fjur	1	29.602	29.602	142,00	208,46
FMas	149	5.297.017	35.550,45	150,42	237,53
FMat	155	5.692.237	36.962,58	151,12	245,13
FNat	776	31.114.537	40.096,05	168,66	239,32
FPhi	134	14.082.919	105.096,40	306,63	336,87
FWir	52	2.469.470	47.489,81	208,21	224,07
Gesamt	1.441	66.923.833	46.752,23	261.537[1]	250,97

Bereits aus Tabelle 4.1 ist die Unausgewogenheit des Korpus zu erkennen. Während nur eine juristische Dissertation vorhanden ist, enthält das Korpus über 776 Dissertationen aus der naturwissenschaftlichen Fakultät. Entsprechend steuern die Naturwissenschaftler_innen auch etwa 50% der Wörter bei.

Schon auf den ersten Blick lassen sich einige Rückschlüsse auf den Umfang und die Form der Dissertationen der einzelnen Fakultäten ziehen. Wirtschaftswissenschaftler_innen bzw. Ma-

[1] Hierbei handelt es sich um die Summe und nicht um den Mittelwert.

thematiker_innen und Physiker_innen schreiben deutlich weniger Seiten (μ = 208,21, bzw. μ = 151,12), um den Doktorgrad zu erlangen, als Architekt_innen (μ = 230,846) und Absolvent_innen der philosophischen Fakultät (μ = 306,634). Allerdings zeigt ein Blick auf die relative Häufigkeitsdichtefunktion (Abbildung 4.1), dass der Unterschied in den Mittelwerten zu einem großen Teil auf einzelnen extrem langen Dissertationen der FArc und FPhi beruht und es große Überschneidungen im Bereich zwischen 150 und 200 Seiten gibt.

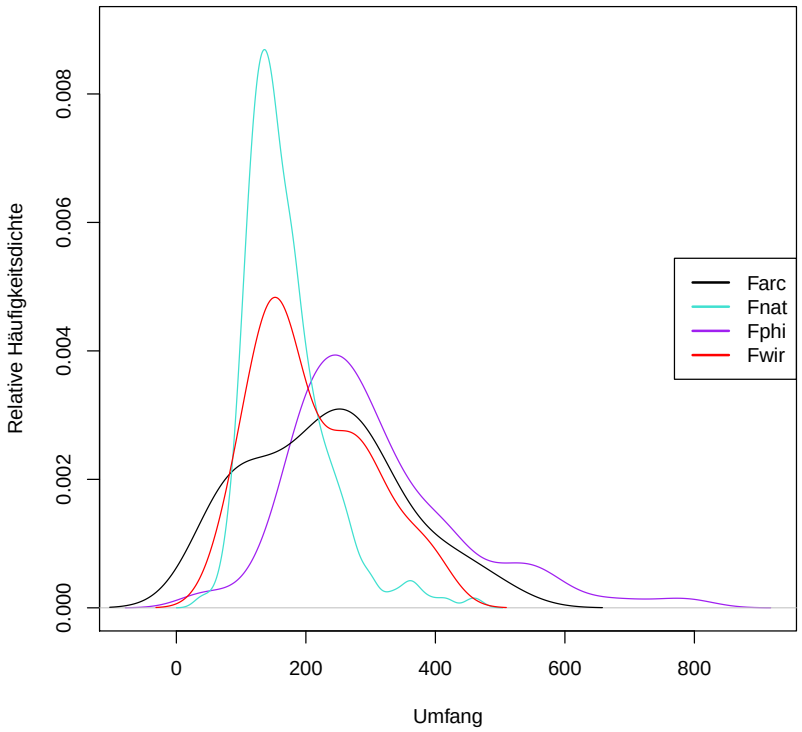


Abbildung 4.1: Relative Häufigkeitsdichtefunktion des Umfangs in Seiten von Dissertationen ausgewählter Fakultäten

Die geringe absolute Quantität an Wörtern geht einher mit einer geringen Dichte an Wörtern pro Seite. Vor allem Formeln, Tabellen und Konstruktionspläne führen dazu, dass bei den Fakultäten für Mathematik und Physik oder für Elektrotechnik und Informatik dreimal weniger Wörter auf eine Seite passen als bei Arbeiten der philosophischen Fakultät. Wirtschaftswissenschaftler_innen befinden sich auf einer Zwischenstufe: Sie schreiben ähnlich lange Dissertationen wie Bauingenieure, bringen allerdings pro Seite etwa 60 Wörter mehr unter.

4.3 Notwendigkeit der Unterscheidung in Sub-Korpora

Abbildung 4.2 zeigt beispielhaft das Problem einer Einteilung der Texte in Sub-Korpora. Auf Basis der Fakultätszugehörigkeit ergibt sich eine Überschneidung aus inhaltlicher als auch sprachlicher Sicht. Einige Texte, wie auch dieser hier, sind interdisziplinär angelegt und nicht trennscharf einer Fakultät zugeordnet.

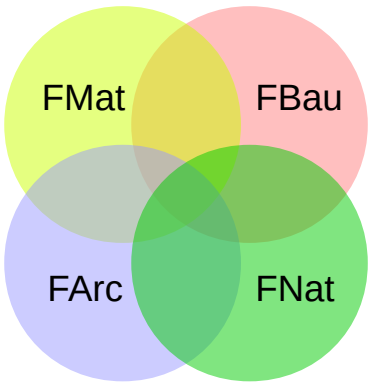


Abbildung 4.2: Beispielhaftes Venn Diagramm für die Fakultäten FMat, FBau, FArc und FNat

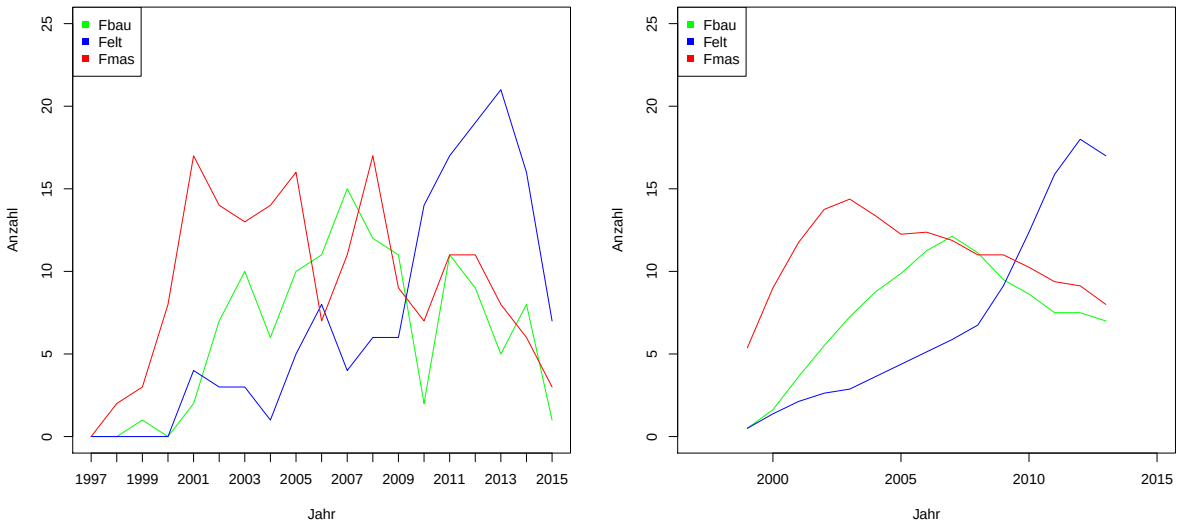
Die folgenden Kapitel werden die Einteilung des Gesamtkorpus in fakultätsspezifische Sub-Korpora beschreiben. Die Einteilung erfolgt entsprechend der Fakultät, welche die Promotion betreut hat. Zunächst soll zwischen dem betrachteten Gegenstand und der Fachwissenschaftssprache unterschieden werden. Der gleiche thematische Gegenstand kann aus mehreren Perspektiven beleuchtet werden. Dies soll am Beispiel eines Hausbaus verdeutlicht werden. Aus Sicht der Fakultät für Architektur wird die optische Gestaltung des Hauses beschrieben, während sich die Bauingenieurwissenschaft etwa mit der Statik der Wände und Decken beschäftigt. Naturwissenschaftler_innen zeigen neue Materialien auf und die Wirtschaftswissenschaft beleuchtet mikroökonomische Finanzierungsmodelle. Auch wenn also das inhaltlich gleiche Thema beschrieben wird, so unterscheiden sich jedoch die Blickwinkel und damit das Vokabular. Kapitel 4.4.3 wird diese sprachwissenschaftliche Unterscheidung mittels Machine Learning Algorithmen empirisch unterstützen. Dennoch bleibt festzuhalten, dass eine scharfe Trennung der einzelnen Fakultäten kaum möglich ist.

4.4 Fokusanalyse der Texte der ingenieurwissenschaftlichen Fakultäten

Obwohl HAAWC Texte aus allen neun Fakultäten der LUH enthält, sind in dieser Arbeit die deutschsprachigen Texte der Ingenieurwissenschaften von besonderem Interesse (siehe Kapitel 3).

Abbildung 4.3 zeigt die Entwicklung der Veröffentlichungszahlen für die Fakultät für Bauingenieurwesen und Geodäsie (FBau), die Fakultät für Elektrotechnik und Informatik (FElt) und die Fakultät für Maschinenbau (FMas) von 1997 bis 2015. Es ist zu beachten, dass das Korpus im Oktober 2015 zusammengestellt wurde und somit nicht alle in diesem Jahr veröffentlichten Dissertationen enthält.

Insgesamt unterliegt die Anzahl der Dissertationen starken Schwankungen. Während die Zahl der deutschsprachigen Dissertationen der FElt stetig steigt, sinken die Zahlen bei den anderen beiden ingenieurwissenschaftlichen Fakultäten seit 2008. Auch bei der FElt ist ein Absinken nach 2013 zu beobachten. Werden die Fluktuationen in den Rohdaten durch Moving



(a) Rohdaten (b) Moving Averages (4 Jahre)

Abbildung 4.3: Anzahl der digital veröffentlichten Dissertationen pro Jahr zwischen 1997 und 2015 für die Fakultäten FBau, FElt und FMas

Averages über 4 Jahre ausgeglichen, so wird der absteigende Trend in allen drei Fakultäten umso deutlicher (Abbildung 4.3b).

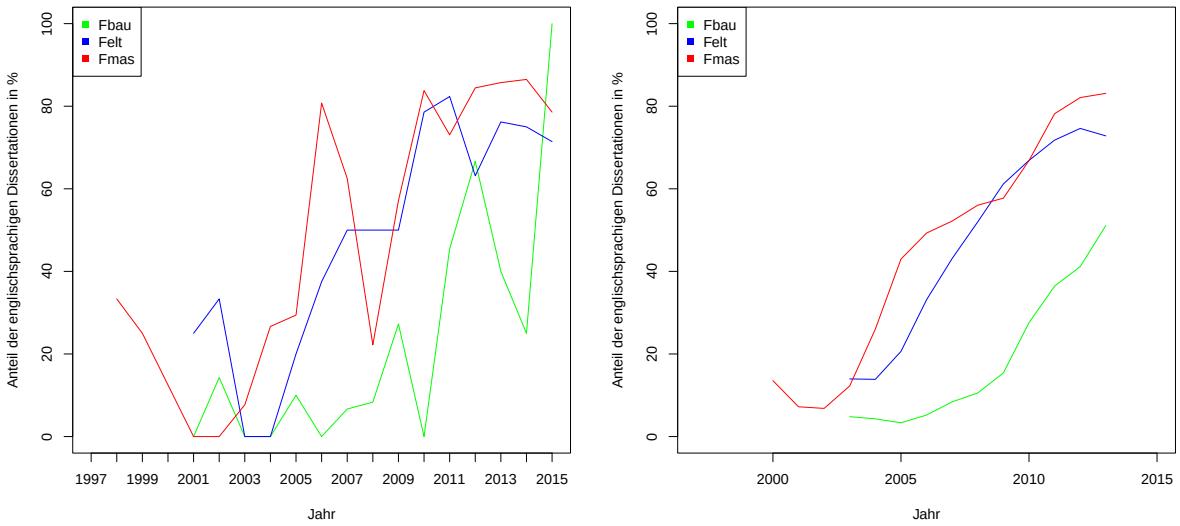
Wird allerdings der Anteil der auf Englisch verfassten Dissertationen betrachtet (siehe Abbildung 4.4), sind zwei klare Trends zu beobachten. Je nach Fakultät werden bis 2003 kaum Dissertationen auf Englisch veröffentlicht. Erst danach steigt der Anteil merklich an. 2008 (FElt) bzw. 2012 (FBau) lag der Anteil der auf Englisch veröffentlichten Dissertationen erstmals über denen, die auf deutsch verfasst wurden. Einzig an der Fakultät für Maschinenbau steigt der Anteil nicht über 36%. Dieser Anstieg erklärt den sinkenden Trend der auf Deutsch verfassten Dissertationen (Abbildung 4.3).

Aus den oben beschriebenen Trends ergeben sich einige Herausforderungen, da die Nachfrage nach Schreibberatungen auf Englisch vermutlich weiter zunehmen wird, wenn Fakultäten nicht mehr auf deutschsprachige Dissertationen und Abschlussarbeiten bestehen. Dementsprechend muss auch die Textgrundlage für HanConc laufend angepasst und parallel auf Englisch aufgebaut werden. Ebenso sollte ein begleitendes Angebot etabliert werden, welches Studierenden ermöglicht, nach einem gegebenenfalls deutschsprachigen Studium einfacher eine englischsprachige Abschlussarbeit zu verfassen.

4.4.1 Methoden zur Terminology Extraction

Die nun folgenden Unterkapitel stellen die drei Sub-Korpora (FBau, FElt und FMas) in ihrem Aufbau und ihren Eckdaten vor[2].

[2]Die Untersuchung mittels einer Terminology Extraction dient der methodischen Vollständigkeit und um zu überprüfen, ob grundsätzlich eine ausreichend große Spezifität des Vokabulars vorhanden ist. Der Fokus dieser Arbeit soll allerdings auf dem Machine Learning Ansatz der folgenden Kapiteln liegen.



(a) Rohdaten

(b) Moving Averages (4 Jahre)

Abbildung 4.4: Anteil der auf Englisch verfassten Dissertation pro Jahr zwischen 1997 und 2015 in Prozent für die Fakultäten FBau, FElt und FMas

Unterschiedliche Fachrichtungen benötigen neben den jeweiligen Fachbegriffen auch ein zugehöriges wissenschaftssprachliches Fachvokabular. Geisteswissenschaftler_innen verwenden zum Beispiel Begriffe wie „diskutieren" und „erörtern" häufiger, als Ingenieurinnen und Naturwissenschaftler_innen, die vermehrt „zeigen", „berechnen" oder „beweisen". Quantitative Nachweise hierfür sollten auch in den entsprechenden Korpora nachweisbar sein. Zuerst soll diese Hypothese mit Hilfe einer klassisch-linguistischen Terminology Extraction bestätigt werden (Kilgarriff 2012). In einem zweiten Schritt wird die Analyse erneut durchgeführt, wobei Algorithmen und Methoden des Supervised Machine Learnings zum Einsatz kommen.

Die Terminology Extraction wird in diesem Fall als Bag-of-Words Analysis mit einem χ^2-Test durchgeführt[3]. Das bedeutet, dass die syntaktische und pragmatische Einbettung nicht beachtet wird, sondern nur die Häufigkeit des Auftretens eines Wortes. Ein χ^2-Test soll sicherstellen, dass die Unterschiede in relativen Häufigkeiten tatsächlich statistisch signifikant sind.

In einem ersten Schritt wird aus allen Korpora eine gemeinsame ausgezählte Gesamtwortliste aus allen Texten erstellt. Fachspezifisch einzigartige Wörter, das heißt Wörter, die nur in einem einzigen Korpus vorkommen, werden gesondert analysiert. Es wird im Anschluss eine Tabelle erstellt, in welche alle gemeinsamen Wörter und die jeweiligen Frequenzen pro Korpus eingetragen werden. Außerdem werden zwei weitere Spalten mit den jeweiligen Wortarten und den Gesamtfrequenzen hinzugefügt.

Der oben beschriebene χ^2-Test wird nun verwendet, um wortartenspezifisch jeweils das Fachvokabular der einzelnen Fakultäten mit dem Gesamtvokabular zu vergleichen. Hierzu wird mit Hilfe des Vier-Feld χ^2-Tests aus Tabelle 4.2 für jedes einzelne Wort und jede Fakultät die Differenz in relativer Häufigkeit, ein χ^2-Wert und ein p-Wert berechnet. Die daraus entstehende

[3]Paquot & Bestgen (2009) weisen darauf hin, dass der hier verwendete χ^2-Test Wörter mit hoher Frequenz bevorzugt und in diese überbewertet.

Tabelle wird nach der Differenz in relativen Häufigkeiten sortiert und für die ingenieurwissenschaftlichen Fakultäten in Kapitel 4.4.5 bis 4.4.7 diskutiert. Um die Lesbarkeit zu wahren, werden die Tabellen für die übrigen Fakultäten unkommentiert in den Anhang verschoben.

Tabelle 4.2: Vier-Feld χ^2-Test zur Terminology Extraction

Gesamtkorpus	Einzelkorpus
Korpusgröße	Korpusgröße
Frequenz	Frequenz

Auf eine vorherige Gewichtung mittels Term Frequency-Inverse Document Frequency (TF-IDF) (Francis & Flynn 2010, Zhao 2013) oder eine Standardisierung mittels z-Score (Bortz 2010) wird bewusst verzichtet, um eine Vergleichbarkeit mit anderen korpuslinguistischen Studien zu gewährleisten (Paquot & Bestgen 2009).

4.4.2 Ergebnisse zur Terminology Extraction

Tabelle 4.3 und Abbildung 4.5 zeigen die deskriptiven Statistiken zur Differenz der relativen Frequenzen der einzelnen Sub-Korpora zum Gesamtkorpus. Auffällig ist, dass sich der größte Teil des Vokabulars in Bezug auf die Effektstärke unauffällig verhält. Sowohl die Tabelle als auch die Boxplots zeigen, dass mindestens die Hälfte aller Verben, Adjektive und Adverbien in ihrer Frequenz unterhalb des messbaren Bereichs vom Gesamtkorpus divergieren. Trotzdem zeigen Kapitel 4.4.3 und 4.4.5 bis 4.4.7, dass es signifikante Unterschiede bezüglich des Vokabulars zwischen den einzelnen Fakultäten gibt. Um die Lesbarkeit der Arbeit zu wahren, werden einige Tabellen und Auswertungen in den Anhang verschoben.

Aus Abbildung 4.5 wird vor allem das Auftreten von fakultätsspezischen Ausreißern deutlich. Während sich die Hälfte aller Differenzen im kaum messbaren Bereich um Null bewegen, gibt es dennoch bei allen Fakultäten augenscheinliche Ausreißer in beide Richtungen. Das Fehlen einer Gewichtung, bzw. einer Standardisierung hat allem Anschein nach nicht dazu geführt, dass, wie auf Basis von Tabelle 4.1 zu erwarten war, vor allem die deutlich längeren Abschlussarbeiten der philosophischen Fakultät und der Fakultät für Wirtschaftswissenschaften viele signifikant zu seltene Wörter aufweisen. Stattdessen zeigt keine der untersuchten Fakultäten eine grundsätzlich auffällige Tendenz in eine Richtung. Mittels des oben beschriebenen Vorgehens konnte somit empirisch gezeigt werden, dass Teile des Vokabulars spezifisch für bestimmte Fakultäten und damit Fachwissenschaften sind.

4.4.3 Klassifikation mittels Supervised Machine Learning

Während sich die Standardwerke der Statistik kaum mit Klassifikationsalgorithmen mit mehr als zwei Klassen beschäftigen (etwa multinominale logistische Regressionen oder multinominale Probit Regressionen (Cameron & Trivedi 2005, Wooldridge 2002)), bietet das Machine Learning dafür eine Reihe von Methoden an. Trotz der unterschiedlichen Logiken hinter den verschiedenen Algorithmen ist die Vorgehensweise doch ähnlich.

Tabelle 4.3: Deskriptive Statistiken der relativen Differenz der Frequenz des nicht-funktionalen Vokabulars (Werte kleiner $|5 \cdot 10^{-5}|$ werden zu null gerundet)

	Minimum	Erstes Quartil	Median	Mittelwert	Drittes Quartil	Maximum
FArc	-0,008	0	0	0	0	0,005
FBau	-0,001	0	0	0	0	0,004
FElt	-0,009	0	0	0,00001	0	0,006
FMas	-0,007	0	0	0	0	0,009
FMat	-0,006	0	0	0	0	0,003
FNat	-0,003	0	0	0	0	0,011
FPhi	-0,017	0	0	0	0	0,004
FWir	-0,014	0	0	0	0,00001	0,004

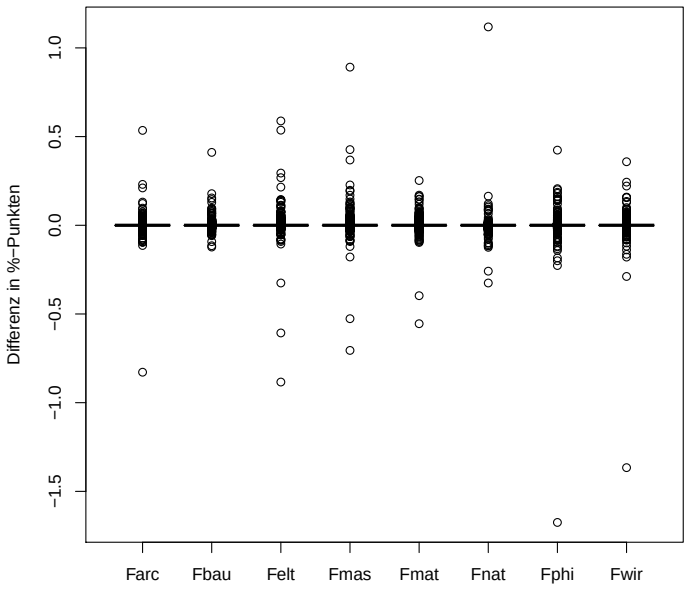


Abbildung 4.5: Boxplots der Differenz in %-Punkten des Vokabulars der einzelnen Sub-Korpora im Vergleich zum Gesamtkorpus

Es wird jeweils eine Matrix erstellt. Diese Matrix besteht wie bei einer Regressionsmatrix aus x_i Spalten mit i Zeilen und einem Vektor y_i mit i Einträgen. Auf Basis von x_i soll nun y_i vorhergesagt werden. Im Gegensatz zu Regressionen handelt es sich bei y nicht um ordinal oder metrisch skalierte Werte, sondern um gleichwertige kategoriale Daten. Um die Güte der einzelnen Modelle zu schätzen, werden die Daten in zwei Teile geteilt: einen Trainingsdatensatz und einen Testdatensatz. Oftmals wird auch, um eine übermäßige Anpassung an den Testdatensatz zu vermeiden, der komplette Datensatz in drei Teile geteilt (Trainings-, Validierungs- und Testdatensatz) oder die Aufteilung mehrfach an zufälligen Stellen durchgeführt (k-fold cross-validation; (Russell & Norvig 2010, Hastie, Tibshirani & Friedman 2009)).

Im Falle dieser Analyse besteht die Matrix aus einer Term-Document Matrix, bei der jedes Wort eine Spalte und jedes Dokument eine Zeile repräsentiert. Jedes Wort muss mindestens in

5% der Texte d.h. Dokumente eines Sub-Korpus und zehn Mal absolut vorkommen, um den Einfluss von $HapaxLegomena$, die Größe der Matrix und die Trainingszeit der Algorithmen zu reduzieren. Die Anzahl an Dissertationen aus der naturwissenschaftlichen Fakultät wurde willkürlich auf 197 gesenkt, um ein Ungleichgewicht der Matrix zu vermeiden. Als zu bestimmende Kategorie wird die Zugehörigkeit zu einer Fakultät gewählt.

Vorgehen:

Die Klassifikation wird mit KNIME (Version 3.2.1) und Weka (Version 3.7) durchgeführt. Hierzu wird die Term-Dokumenten Matrix (TDM) aus R als ARRF Datei exportiert und in KNIME importiert[4]. Der Datensatz wird dann in einen Training- und einen Testdatensatz aufgeteilt. Der Trainingsdatensatz entspricht dabei zufällig gezogenen 75% des Originaldatensatzes.

Die Vielzahl an Klassifizierungsalgorithmen sollen in zwei Dimensionen kategorisiert werden: Vorhersagegenauigkeit und Erklärungsgehalt. Die Vorhersagegenauigkeit soll hierbei für die Genauigkeit des Machine Learning Algorithmus' stehen. Hierfür wird der Algorithmus zuerst auf die Trainingsdaten angewendet und das sich daraus ergebende Modell auf die Testdaten angewandt. Die vom Modell vorgeschlagenen Kategorien werden dann mit den tatsächlichen Kategorien des Testdatensatzes verglichen. Die dabei generierte Matrix wird Confusion Matrix genannt (Vitartas, Heath, Midford, Ong, Alahakoon & Sullivan-Mort 2016). Genauere Modelle kategorisieren hier einen höheren Anteil korrekt.

Abhängig von der Fragestellung kann entweder die Vorhersagegenauigkeit oder der Erklärungsgehalt von Modellen wichtiger sein. Bei geschlossenen Fragestellungen wie Spamerkennung ist es irrelevant, welche Wörter dazu führen, dass eine E-Mail als Spam klassifiziert wird. Stattdessen ist es wichtiger, dass den Nutzer_innen möglichst wenig Spam angezeigt wird und gleichzeitig möglichst keine erwünschten Mails als Spam kategorisiert und gelöscht werden. Bei offenen Fragestellungen wie nach der Kategorisierbarkeit von Dissertationen geht es weniger darum, die Fakultätszugehörigkeit möglichst genau vorherzusagen. Stattdessen sollen die Ergebnisse und vor allem die Zwischenschritte für den Menschen lesbar sein.

Abbildung 4.6 verdeutlicht noch einmal exemplarisch das oben beschriebene Vorgehen. Der Naïve Bayes Classifier und die auf Entscheidungsbäumen basierenden Algorithmen werden mit den voreingestellten Einstellungen verwendet. Beide neuronale Netzwerke bestehen jeweils aus acht versteckten Schichten und zehn Neuronen pro Schicht.

Abbildung 4.7 zeigt die Vorhersagegenauigkeit der eingesetzten Klassifikationsalgorithmen. Es fällt auf, dass die auf neuronalen Netzwerken beruhenden Algorithmen deutlich ungenauer als die auf Entscheidungsbäumen basierenden sind. Selbst ein einfacher Naïve Bayes Classifier hat eine deutlich höhere Präzision. Wird etwa beim Multilayer Perceptron die Anzahl der Neuronen pro Schicht auf 20 erhöht, so steigt zwar die Präzision auf 63%, allerdings steigt ebenso die Laufzeit deutlich an. Um die Werte der Entscheidungsbäume zu erreichen, wären deutlich mehr Neuronen und Schichten notwendig, wobei dann, aufgrund der geringen Anzahl an Dissertationen im Vergleich zu den eingesetzten Wörtern, die Gefahr einer Überanpassung steigt.

[4]Weka wird über KNIME aufgerufen, um sicherzustellen, dass bei jedem Algorithmus die gleiche Aufteilung in Trainings- und Testdaten stattfindet.

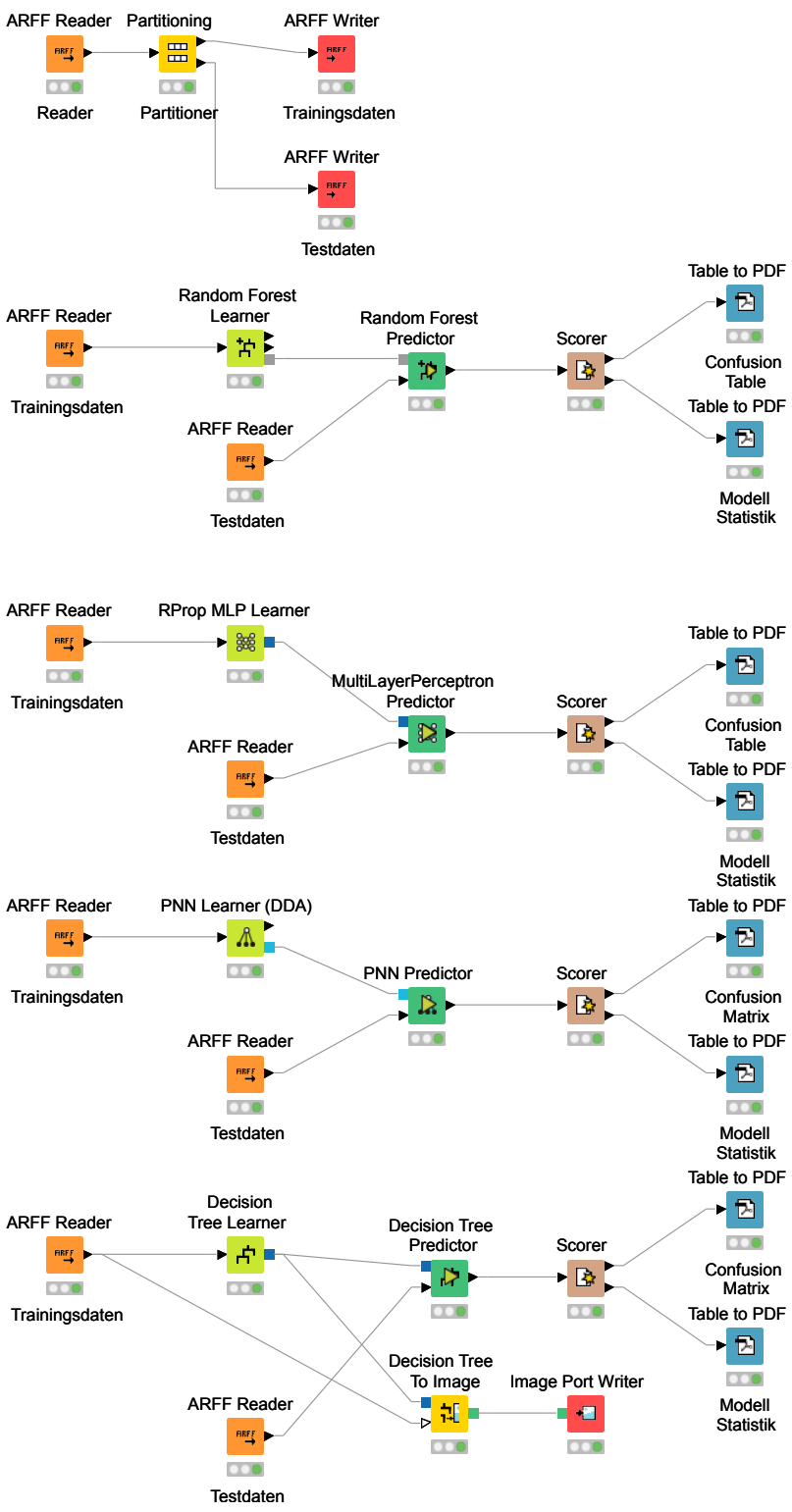


Abbildung 4.6: KNIME Workflow zur Klassifikation von Texten nach Fakultäten mit ausgewählten Algorithmen

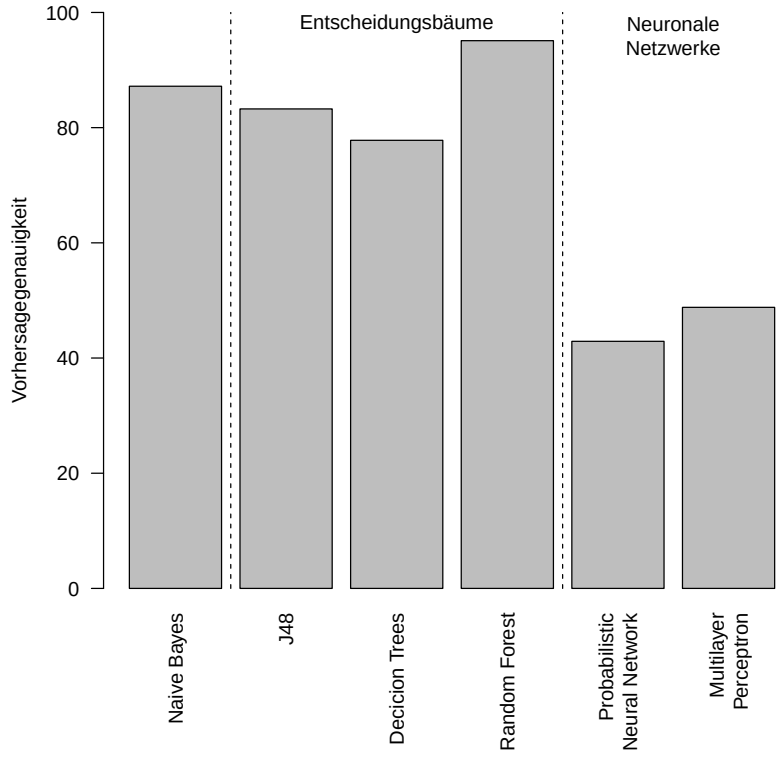


Abbildung 4.7: Vorhersagegenauigkeit der Fakultätszugehörigkeit von Dissertationen von verschiedenen Klassifikationsalgorithmen in %

Interpretierbarkeit der Algorithmen

Wie oben bereits angedeutet, unterscheiden sich die Algorithmen unter anderem dadurch, wie einfach ihre Zwischenschritte und Ergebnisse zu interpretieren sind. Die Zwischenschritte Neuronaler Netzwerke und des Random Forest Algorithmus sind für den Menschen nicht zu interpretieren; die des Decision Trees, von J48 und des Naïve Bayes Classifiers hingegen schon. Bei Decision Trees und J48 kann ein Entscheidungsbaum geplottet werden. Die Ergebnisse von Naïve Bayes Classifiern können als Skatterplots mit der Frequenz eines Wortes auf der x-Achse und der Wahrscheinlichkeitsdichte der einzelnen Fakultäten auf der y-Achse dargestellt werden. Die Darstellung mit Scatterplots hat allerdings den Nachteil, dass sie im Fall dieser Arbeit zu über 44.000 Plots führen würde.

Ergebnisse

Der J48 und der Decision Tree Algorithmus klassifizieren ausreichend genau (siehe Abbildung 4.7) und sind einfach zu interpretieren. Vom Wurzelknoten (root node) aus kann ein Text aufgrund der Frequenz einzelner Wörter einer Fakultät zugeordnet werden. Jeder Knoten entspricht dabei einer Entscheidung. Je nach Frequenz der einzelnen Wörter im zu klassifizierenden Text wird im Baum in der darunter befindlichen Ebene nach links oder rechts weitergegangen. An den Enden befindet sich jeweils ein Blatt (leaf), das die zugeordnete Fakultät und die Frequenz der zugeordneten Texte angibt.

Tabelle 4.4 zeigt die Vorhersagegenauigkeit im Detail. Jede Spalte steht hierbei für die vom

Algorithmus vorhergesagte Fakultät, während jede Zeile die tatsächliche Fakultät zeigt. Bei einem perfekt klassifizierenden Modell würde an jeder Kreuzung von gleicher Fakultät in Zeile und Spalte die genaue Anzahl der Texte der jeweiligen Fakultät im Testdatensatz stehen.

Tabelle 4.4: Confusion Matrix der Klassifikationergebnisse des J48 Algorithmus' mit vorhergesagten Fakultäten in den Spalten und tatsächlichen Fakultäten in den Zeilen

Fakultät	FBau	FElt	FMas	FMat	FNat	FPhi	FWir
FBau	14	0	3	0	1	0	0
FElt	2	3	1	0	5	0	0
FMas	5	0	38	0	0	1	0
FMat	0	0	1	32	2	1	0
FNat	1	0	0	0	44	2	0
FPhi	2	2	1	0	2	27	1
FWir	0	1	0	0	0	0	11

Abbildung 4.8 zeigt die Baumstruktur der J48 Klassifikation. Dieser Baum beschreibt, welche Wörter maßgeblich für die Vorhersagegenauigkeit sind. Die weißen Ovale stehen jeweils für die einzelnen Knoten und die grauen Kästen für die Blätter. Die Frequenzen an den Kanten zwischen den Knoten entsprechen den absoluten Frequenzgrenzen in den Dissertationen. Von den über 44.000 Wörtern der Matrix bleiben 32 übrig, um eine Dissertation mit einer Wahrscheinlichkeit von 83,25% korrekt zu klassifizieren.

Um unterschiedliche grammatikalisch begründete Schreibweisen einzelner Wörter zusammenzufassen, wurden Großbuchstaben durch Kleinbuchstaben ersetzt[5] und mittels eines Stemmers (SnowballC Version 0.5.1) grammatikalische Suffixe[6] entfernt. An sechs Stellen des Entscheidungsbaumes befinden sich Artefakte (gelb markiert), die noch aus der Konvertierung der PDFs zu TXT entstammen.

Auffällig ist der hohe Grad an binären Entscheidungen (grün markiert). Dadurch, dass so gut wie alle Entscheidungen binär getroffen werden, wird die fachrichtungsspezifische Exklusivität des Vokabulars aufgezeigt. Bei diesem handelt es sich nicht ausschließlich um Fachtermini, sondern auch Begriffe wie zum Beispiel „verdanken" und „auslegen".

Die trainierten Modelle sind ausreichend genau, um Texte ihren Fakultäten zuzuordnen. Im Umkehrschluss bedeutet dies, dass die Ursprungsfakultät ein ausreichendes Kriterium zur Organisation der Korpora ist. Die folgenden Kapitel zeigen auf Basis von linguistischen Kennzahlen die sprachlichen Unterschiede zwischen den Dissertationen der verschiedenen Fakultäten.

4.4.4 Type-Token Ratio (TTR) Wachstumskurve

Als Maß für sprachliche Variabilität wird vielfach die Type-Token Ratio herangezogen (Verspoor, Lowie, Dijk & Van Dijk 2008, Vyatkina 2012, Friginal & Weigle 2014, Biber 1992, De Haan & Van der Haagen 2013, Grotjahn 2002, Louwerse et al. 2003). Dabei wird die Anzahl der

[5]Hierdurch wird der Einfluss der Satzstellung und der Satzzeichen entfernt.

[6]Damit sind Pluralendungen, Flexionen bei Verben und Steigerungsformen bei Adjektiven gemeint.

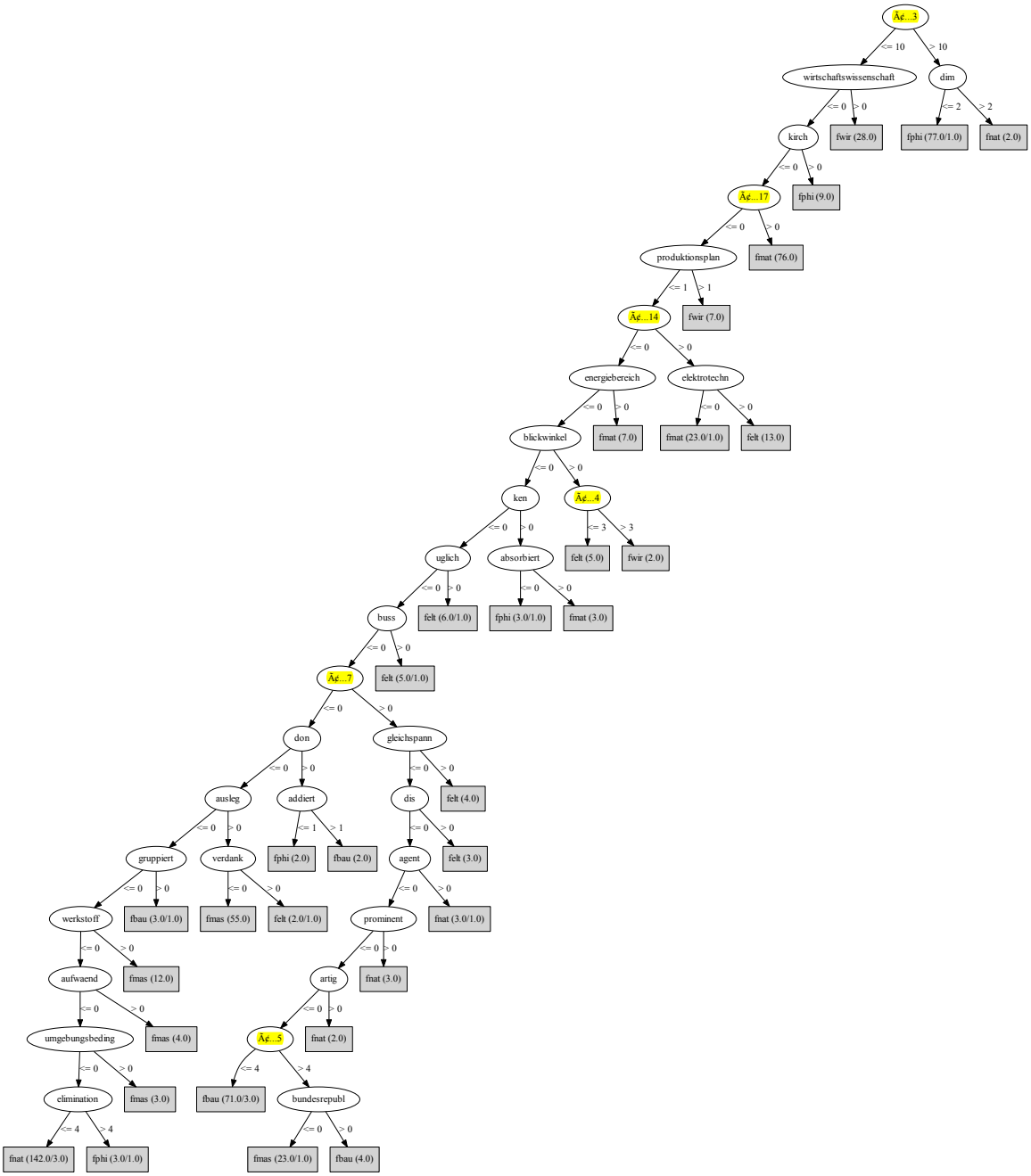


Abbildung 4.8: Entscheidungsbaum des J48 Algorithmus' zur Klassifikation der Dissertationen nach Fakultät mit farblich markierten Artefakten

einzigartigen Wörter (Types) durch die Gesamtzahl der Wörter geteilt (Token/Korpusgröße)[7], sodass:

$$TTR = \frac{n_{Type}}{n_{Token}} \tag{4.1}$$

Allerdings weisen verschiedene Autor_innen darauf hin, dass die TTR hochgradig abhängig von der Korpusgröße (Baayen 2001, Durrant & Schmitt 2009) und der Homogenität der Texte ist (McCarthy & Jarvis 2010). Vor allem bei kurzen Texten, wie sie etwa in De Haan & Van der Haagen (2013) analysiert werden, ist die Anfälligkeit für statistische Artefakte besonders hoch (Mckee, Malvern & Richards 2000).

Durrant & Schmitt (2009) schlagen als Alternative eine Median TTR pro 100 Tokens vor. McCarthy & Jarvis (2010) weisen in diesem Zusammenhang darauf hin, dass eine so kleine Einteilung zu einer höheren Sensitivität führt. Die Alternativen voc-D und HD-D werden mit Verweis auf McCarthy & Jarvis (2010) aus dem gleichen Grund ebenfalls verworfen.

McCarthy & Jarvis (2010) diskutieren die Measure of Textual Lexical Diversity (MTLD) als besseres Maß für lexikalische Diversität. Hierbei wird die mittlere Anzahl an Tokens zurückgegeben, die nötig ist, um die TTR unter 0,72 zu drücken. Allerdings geben McCarthy & Jarvis (2010) zu bedenken, dass die MTLD noch nicht komplett erforscht sei. Deshalb wird an dieser Stelle auf den Einsatz verzichtet.

Um dennoch den berechtigten Anmerkungen bezüglich des Einflusses der Korpusgröße Rechnung zu tragen, werden statt eines einzelnen TTR Wertes TTR Wachstumskurven benutzt, um die lexikalische Variabilität der einzelnen Sub-Korpora aufzuzeigen. Die Wachstumskurven werden wortartenspezifisch ausgewertet und dargestellt, um zu verhindern, dass etwa die Größe des Fachvokabulars eine höhere lexikalische Diversität vortäuscht. Dementsprechend werden in den folgenden Abbildungen die Wachstumskurven für Verben, Adjektive und Adverbien geplottet[8]. Dabei beziehen sich die Types und Tokens jeweils nur auf die entsprechenden Wortarten.

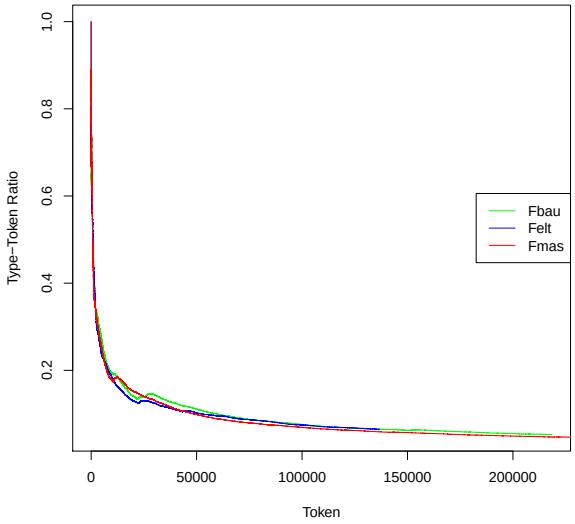
Abbildung 4.9a zeigt die TTR für die Verben aus den drei ingenieurwissenschaftlichen Fakultäten in Abhängigkeit von der Korpusgröße. Auffällig ist, dass die TTR für alle drei Fakultäten exponentiell sinken und asymptotisch gegen 0,05 auslaufen. Nur in einem Bereich von 10.000 bis 60.000 sind merkliche Unterschiede festzustellen. Die TTR scheint daher ungeeignet, um die sprachliche Variabilität einzelner Fakultäten abzudecken.

Die unterschiedliche Länge der Kurven in Abbildung 4.9 liegt in der Korpusgröße begründet. Aufgrund von Überschneidungen sind einige Bereiche überlagert. Die Unterschiede zwischen den einzelnen Kurven sind mit maximal 0,03 verschwindend gering.

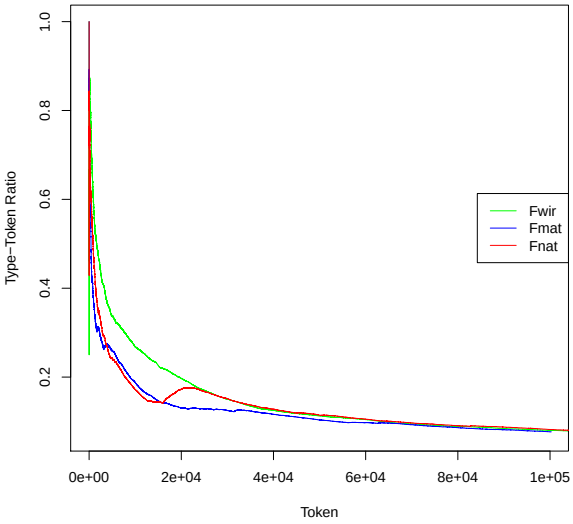
Aus Abbildung 4.9b wird die Sensitivität der TTR in Bezug auf persönlichen Schreibstil deutlich. Bis zu Token 15.000 verhalten sich die Wachstumskurven wie erwartet. Die beiden mathematiklastigen Fakultäten (FMat und FNat) haben eine deutlich niedrigere TTR als die

[7] Der Satz „Der Baum, der dort auf der Wiese steht, ist der schönste Baum im Sommer." besteht beispielsweise aus 14 Token aber nur aus 10 Types, da die Wörter „der" und „Baum" insgesamt drei bzw. zwei mal verwendet werden.

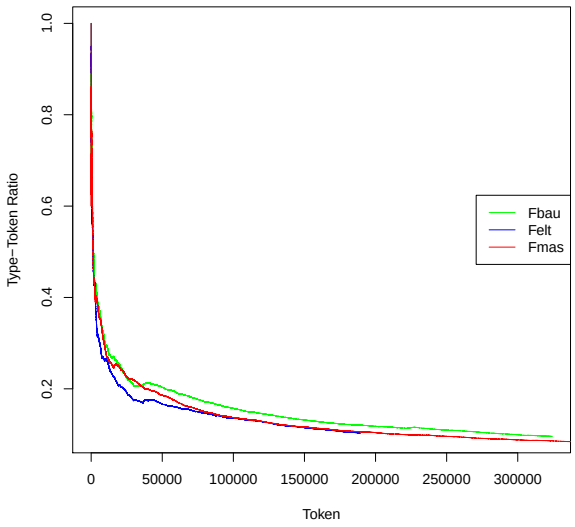
[8] Auf die übrigen Wortarten wurde verzichtet, weil sie entweder syntaktische Funktionen haben (etwa Pronomina) oder eine geschlossene Gruppe sind (etwa Konjunktionen).



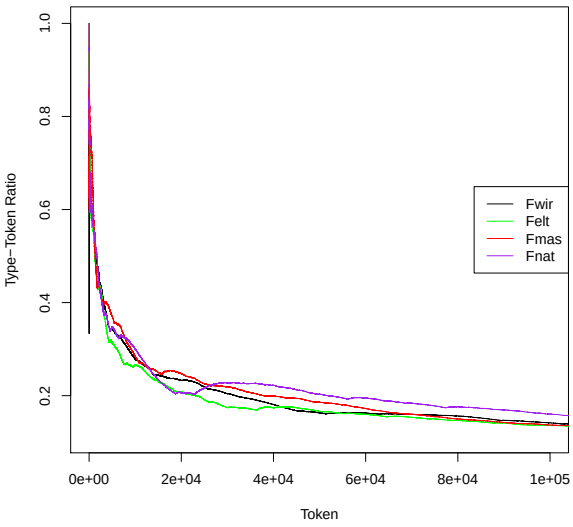
(a) FBau/FElt/FMas (Verben)



(b) FWir/FMat/FNat (Verben)



(c) FBau/FElt/FMas (Adjektive)



(d) FWir/FElt/FMas/FNat (Adjektive)

Abbildung 4.9: Type-Token Ratio (Verben (a-b) & Adjektive (c-d)) in Abhängigkeit von der Anzahl an Tokens für jeweils mehrere Fakultäten

eher geisteswissenschaftlich orientierte FWir.

Die Abbildungen 4.9c und 4.9d weisen analog zu Abbildung 4.9a und 4.9b die Wachstumskurven der Adjektive aus. Auch hier sterben die Grafiken jeweils gegen 0,01 und 0,2. Die Unterschiede zwischen den einzelnen Fakultäten sind kaum ausgeprägter als bei den Verben. Auffällig ist jedoch, dass die Unterschiede vielfach nicht stetig sind, sondern im Verlauf der Kurven mehrfach die Richtung wechseln.

Die Unterschiede bei Adverbien sind verschwindend gering. Deshalb wird an dieser Stelle auf sie verzichtet und auf den Anhang verwiesen.

Die folgenden Kapitel zeigen die deskriptiven Kennwerte und die detaillierten Ergebnisse aus der Terminology Extraction und der TTR Analyse für die einzelnen ingenieurwissenschaft-

lichen Sub-Korpora. Die Ergebnisse der restlichen Sub-Korpora befinden sich im Anhang.

4.4.5 Dissertationen der Fakultät für Bauingenieurwesen und Geodäsie

An der Fakultät für Bauingenieurwesen und Geodäsie (FBau) wurden seit 1997 insgesamt 121 Dissertationen digital als PDF veröffentlicht. Von diesen 121 Dissertationen wurden 98 (80,99%) auf Deutsch geschrieben. Die deutschsprachigen Texte summieren sich auf 17.926 Seiten bei im Schnitt 182,9 Seiten pro Arbeit ($\sigma = 81{,}93$) und 4.112.603 Wörtern ($\mu = 41.965{,}34$, $\sigma = 22.037{,}61$), während die englischsprachigen Dissertationen nur auf 3.777 Seiten ($\mu = 164{,}22$, $\sigma = 62{,}17$) kommen[9].

Abbildung 4.10 zeigt die Wachstumskurven für die Verben, Adjektive und Adverbien im deutschsprachigen Teil des FBau Sub-Korpus. Wie in Kapitel 4.4.4 beschrieben, wird auf eine Darstellung einer allgemeinen TTR verzichtet, um eine Verzerrung durch Fachvokabular zu vermeiden. Die Länge der einzelnen Kurven entspricht der Korpusgröße der entsprechenden Wortart. Es wird deutlich, dass weniger Adverbien als Verben und weniger Verben als Adjektive verwendet werden. Dieses Verhalten bezieht sich sowohl auf die Anzahl als auch auf die Variabilität der jeweiligen Wortarten[10].

Tabelle 4.5[11] zeigt die signifikant abweichenden Nutzungen von Verben und Adjektiven im Vergleich zum Gesamtkorpus. Die signifikant häufiger verwendeten Verben und Adjektive passen zum zu erforschenden Gegenstand der Dissertationen der FBau, denn Bauwerke und ihre Bestandteile werden eher „berechnet" und Eigenschaften „gemessen" oder „ermittelt". Weniger beschäftigen sich Bauingenieur_innen mit „elektrischen" oder „deutschen", „politischen" oder „sozialen" Dingen. Auch wird wenig „beobachtet". Insgesamt zeigt sich vor allem eine Abweichung durch Negativselektion zur Elektrotechnik, Statistik und Geisteswissenschaft. Die Abweichungen in Bezug auf Adverbien und Konjunktionen sind unauffällig und befinden sich als Tabelle 6.6 im Anhang.

4.4.6 Dissertationen der Fakultät für Elektrotechnik und Informatik

Seit 2001 wurden an der Fakultät für Elektrotechnik und Informatik (FElt) insgesamt 134 Dissertationen in digitaler Form eingereicht. Vor allem in den letzten Jahren wurden vermehrt Dissertationen auf Englisch geschrieben. Seit 2010 wurden im Verhältnis zu jeder deutschsprachige etwa zwei bis drei englischsprachige Arbeiten eingereicht. Es ist davon auszugehen, dass dieser Trend anhalten wird und damit die Basis für ein aktuelles deutschsprachiges Korpus der Elektrotechnik und Informatik kleiner werden wird. Im Vergleich zur FBau und FMas werden an der FElt im Durchschnitt mehr Worte insgesamt und pro Seite benötigt, um einen Doktortitel zu erlangen. Die Wachstumskurven der TTR ähneln denen der beiden anderen ingenieurwissenschaftlichen Fakultäten und werden daher hier nicht dargestellt. Die Beobachtungen zu den

[9] Weil die englischsprachigen Dissertationen nicht im Fokus dieser Arbeit stehen, sind auch keine weiteren Kennzahlen berechnet worden.

[10] Aufgrund der hohen Ähnlichkeit der Werte bei den übrigen Fakultäten wurde auf diese Grafik verzichtet.

[11] Auf die korrespondierenden p-Werte wird hier verzichtet, da sie allesamt kleiner als 0,00001 sind.

Tabelle 4.5: Signifikant häufiger bzw. seltener an der FBau benutzte Verben und Adjektive jeweils mit χ^2-Wert und Differenz zum Gesamtkorpus

Verb	χ^2-Wert	Differenz	Adjektiv	χ^2-Wert	Differenz
verwendet	290,61	0,00018	numerischen	1.415,88	0,00012
durchgefuehrt	287,48	0,00017	waehrend	91,64	0,00011
koennen	95,83	0,00017	raeumlichen	883,73	0,00011
berechnet	617,35	0,00016	moeglich	111,15	0,00010
dargestellt	252	0,00015	maximalen	617,65	0,00010
erfolgt	239,17	0,00014	entsprechend	235,50	0,00010
liegen	209,65	0,00011	maximale	401,28	0,00009
ermittelt	257,21	0,00010	ueber	19,20	0,00009
betraegt	302,52	0,00010	gemessenen	231,83	0,00008
erreicht	169,72	0,00010	unterschiedlichen	84,93	0,00008
⋮	⋮	⋮	⋮	⋮	⋮
lassen	27,95	-0,00004	neue	34,39	-0,00004
machen	69,59	-0,00005	elektrischen	119,05	-0,00004
kommt	36,71	-0,00005	deutschen	69,79	-0,00004
gezeigt	36,17	-0,00005	andere	35,70	-0,00005
zeigen	34,28	-0,00005	eigene	117,37	-0,00005
scheint	130,97	-0,00005	signifikant	80,44	-0,00005
beobachtet	67,81	-0,00006	eigenen	82,74	-0,00005
erfolgte	43,15	-0,00006	anderen	21,45	-0,00005
zeigten	92,65	-0,00006	politischen	240,06	-0,00006
koennte	128,88	-0,00009	sozialen	283,52	-0,00009

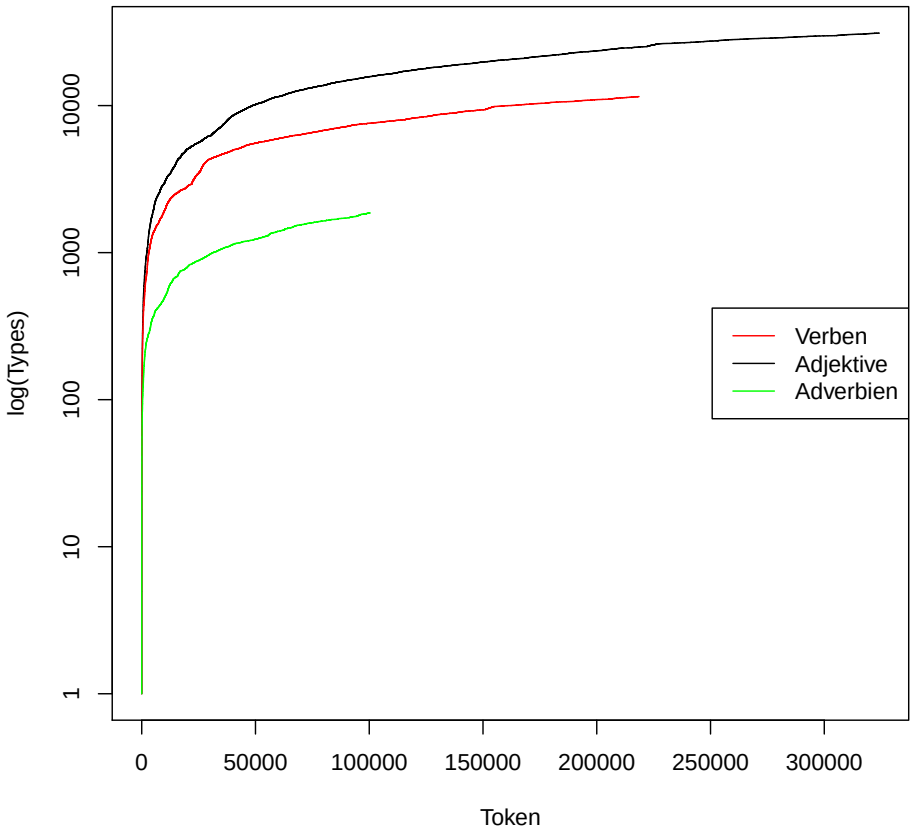


Abbildung 4.10: log(Type)-Token Ratio Wachstumskurven für Verben, Adjektive und Adverbien an der FBau

signifikant häufiger oder seltener verwendeten Verben und Adjektiven stimmen mit denen der FBau überein. Einzig das Wort „elektrischen" wird aus offensichtlichen Gründen häufiger als im Gesamtkorpus verwendet.

4.4.7 Dissertationen der Fakultät für Maschinenbau

Der Maschinenbau hat im Gegensatz zur Elektrotechnik und wahrscheinlich vor allem zur Informatik einen deutlich höheren Schwerpunkt im deutschsprachigen Raum. Während an der FElt 50 deutschsprachige bzw. 84 englischsprachige Dissertationen eingereicht wurden, ist das Verhältnis an der FMas entgegengesetzt. Hier wurden seit 1998 149 Dissertationen auf Deutsch eingereicht, während es im gleichen Zeitraum nur 28 auf Englisch waren. Über die Gründe soll hier nicht spekuliert werden.

Insgesamt ist das FMas Korpus das größte aller drei ingenieurwissenschaftlichen Korpora. Auch wenn die absolute Anzahl an Wörtern hier am höchsten ist, sind die einzelnen Dissertationen dennoch etwa 7.000 Wörter und 26 Seiten kürzer als bei den anderen beiden Fakultäten.

In Bezug auf signifikant anders verwendete Verben und Adjektive zeigt die FMas (siehe Tabelle 4.7), dass häufig „experimentell[...]" an „mechanischen" Dingen „gemessen" und „be-

Tabelle 4.6: Signifikant häufiger bzw. seltener an der FElt benutzte Verben und Adjektive jeweils mit χ^2-Wert und Differenz zum Gesamtkorpus

Verb	χ^2-Wert	Differenz	Adjektiv	χ^2-Wert	Differenz
koennen	$987,02$	$0,00074$	moeglich	$414,62$	$0,00027$
ergibt	$1.412,53$	$0,00042$	einzelnen	$368,63$	$0,00024$
verwendet	$616,05$	$0,00036$	ueber	$49,33$	$0,00020$
berechnet	$606,87$	$0,00022$	elektrischen	$801,44$	$0,00015$
gilt	411	$0,00021$	realen	$939,58$	$0,00014$
beruecksichtigt	747	$0,00021$	notwendig	$289,96$	$0,00014$
muessen	$384,70$	$0,00020$	folgenden	$173,24$	$0,00013$
entspricht	$411,17$	$0,00019$	verschiedene	$161,40$	$0,00013$
betrachtet	$502,19$	$0,00018$	beschriebenen	$265,63$	$0,00012$
beschrieben	$210,30$	$0,00017$	geschaetzten	$1.935,43$	$0,00012$
⋮	⋮	⋮	⋮	⋮	⋮
koennte	$37,41$	-$0,00006$	Deutschen	$84,75$	-$0,00005$
zeigen	$30,77$	-$0,00007$	spezifische	$92,91$	-$0,00006$
geloest	$97,93$	-$0,00007$	optischen	$108,56$	-$0,00006$
fuehrte	$131,65$	-$0,00008$	politischen	$134,41$	-$0,00007$
versetzt	$151,13$	-$0,00008$	spezifischen	$104,72$	-$0,00007$
getrocknet	$175,47$	-$0,00008$	Anschliessend	$81,81$	-$0,00007$
nachgewiesen	$121,60$	-$0,00009$	deutschen	$144,60$	-$0,00008$
zeigten	$142,89$	-$0,00010$	organischen	$171,28$	-$0,00009$
zeigte	$132,13$	-$0,00010$	signifikant	$132,02$	-$0,00009$
erfolgte	$327,91$	-$0,00021$	anschliessend	$105,36$	-$0,00010$

rechnet" wird und die Ergebnisse „dargestellt" werden. Bei den signifikant seltener verwendeten Verben und Adjektiven fallen vor allem neben der Abgrenzung zu den eher geisteswissenschaftlichen Begriffen („Deutschen", „politischen" und „sozialen") die fehlenden Begriffe aus dem Bereich der Chemie auf. So wird seltener etwas „getrocknet" oder mit „organischen" Dingen „versetzt". An dieser Stelle sei auf die zusätzliche Tabelle 6.8 im Anhang verwiesen. Bei den dort dargestellten signifikant häufiger verwendeten Adverbien fallen „links" und „rechts" besonders auf. Hier liegt die Vermutung nahe, dass diese vor allem für die Beschreibung von Konstruktionsplänen verwendet werden.

Tabelle 4.7: Signifikant häufiger bzw. seltener an der FMas benutzte Verben und Adjektive jeweils mit χ^2-Wert und Differenz zum Gesamtkorpus

Verb	χ^2-Wert	Differenz	Adjektiv	χ^2-Wert	Differenz
koennen	$604,98$	$0,00039$	hohen	$1.434,94$	$0,00028$
ergibt	$2.025,19$	$0,00035$	waehrend	$364,86$	$0,00021$
erfolgt	$1.561,56$	$0,00033$	deutlich	$383,52$	$0,00019$
dargestellt	$1.237,65$	$0,00030$	gemessenen	$1.288,92$	$0,00018$
zeigt	$777,96$	$0,00028$	hohe	$482,26$	$0,00016$
fuehrt	$787,99$	$0,00023$	experimentellen	$1.453,64$	$0,00015$
berechnet	$1.364,64$	$0,00023$	berechneten	$1.446,30$	$0,00015$
bestimmt	$490,88$	$0,00018$	mechanischen	$2.066,88$	$0,00015$
erreicht	$671,67$	$0,00018$	moeglich	238	$0,00014$
laesst	$370,28$	$0,00017$	unterschiedlichen	$297,60$	$0,00013$
⋮	⋮	⋮	⋮	⋮	⋮
lag	$241,75$	-$0,00007$	Deutschen	$179,79$	-$0,00005$
machen	$176,27$	-$0,00007$	organischen	$157,26$	-$0,00006$
waere	$152,88$	-$0,00007$	eigene	$214,42$	-$0,00006$
erhalten	$101,78$	-$0,00007$	signifikant	$175,77$	-$0,00007$
versetzt	$263,83$	-$0,00007$	politischen	$328,04$	-$0,00007$
getrocknet	$311,89$	-$0,00007$	andere	$104,30$	-$0,00007$
zeigten	$224,45$	-$0,00008$	deutschen	$321,61$	-$0,00008$
zeigte	$243,70$	-$0,00009$	eigenen	$413,95$	-$0,00011$
koennte	$478,97$	-$0,00015$	sozialen	$550,70$	-$0,00011$
erfolgte	$503,70$	-$0,00017$	anderen	$132,74$	-$0,00012$

4.5 Zwischenfazit

Die vorangegangen Kapitel haben gezeigt, dass Dissertationen eine geeignete Textgrundlage für eine korpusgestützte akademische Schreibberatung sind. Entsprechend der Struktur der LUH wurde überprüft, wie ein solches Korpus designt werden muss. Ausgehend von der Hypothese, dass es gravierende sprachliche Unterschiede zwischen einzelnen Wissenschaftstraditionen gibt, wurde ein Modell entwickelt, um zu belegen, dass eine Einteilung in Subkorpora sinnvoll ist. Entsprechend dem quantitativen Vorgehen dieser Arbeit wurden verschiedene Methoden aus der Sprachwissenschaft, Statistik und des Machine Learnings verwendet, um eine Eintei-

lung des Gesamtkorpus in Subkorpora nach Fakultätszugehörigkeit als ein sinnvolles Kriterium empirisch nachzuweisen.

Die Untersuchung der einzelnen Korpora hat ergeben, dass die sprachlichen Unterschiede zwischen den einzelnen Fakultäten statistisch signifikant sind. Schreibberater_innen müssen sich daher detaillierter mit einzelnen sprachlichen Konstruktionen in den Arbeiten ihrer Studierenden auseinandersetzen, um sicher zu gehen, dass diese Konstruktionen dem Sprachgebrauch der Zielfakultät entsprechen. Aus dieser Erkenntnis folgt, dass tiefergehende Funktionalitäten, welche den typischen Sprachgebrauch besser beschreiben als nur einfache KWIC, implementiert werden müssen. Erst durch solche Möglichkeiten werden Schreibberater_innen in die Lage versetzt, zu verstehen, wie die einzelnen Fakultäten schreiben und nicht nur worüber, um somit ihre Studierenden auch besser beraten können.

Kapitel 5

HanConc

HanConc (Hannover Concordancer) ist als offene Plattform zur linguistischen Analyse von wissenschaftlichen Texten konzipiert. Das Hauptaugenmerk beim Design der Plattform liegt auf einer hohen Flexibilität und Anpassbarkeit an unterschiedliche Zielgruppen (siehe auch Kapitel 3). Die hier vorgestellte Version von HanConc ist an die Bedürfnisse von Schreibberater_innen angepasst. Um HanConc auch für andere Zielgruppen oder aber bei sich verändernden Anforderungen anpassen zu können, ist der Quellcode offen und möglichst einfach programmiert. Sowohl das Frontend als auch das Backend sind in R programmiert. Obwohl HyperText Markup Language (HTML) und Cascading Style Sheets (CSS) für die Gestaltung des Frontends benutzt wurden, sind keine Kenntnisse erforderlich, um HanConc an die speziellen Herausforderungen der verschiedenen Schreibzentren oder sogar einzelner Schreibberater_innen anzupassen.

Aus drei Gründen wurde R für die Programmierung verwendet (R Core Team 2021):

Erstens ist die Programmiersprache unter bestimmten Bedingungen sehr performant. Alle Daten werden intern in R gespeichert und im RAM gehalten. Potentiell längere Laufzeiten durch die Kommunikation mit einer Datenbank, welche eventuell die Daten erst von einer Festplatte lesen muss, können somit verhindert werden. Da der gesamte Prozess in R gehalten ist, müssen die Daten auch nicht in einem allgemein lesbaren Format vorliegen.

Zweitens ist R eine vielen Linguist_innen bereits bekannte Sprache (Gries 2009, Baayen 2001, Baayen 2008, Evert & Baroni 2007). Dies führt dazu, dass die potentielle Zielgruppe Weiterentwicklungen ohne externe Programmierer_innen durchführen kann. Es gibt jedoch auch viele andere Pakete für korpus- und computer-linguistische Analysen auf Basis verschiedenster Programmiersprachen. Mit NLTK für die linguistischen Analysen (Bird, Klein & Loper 2009) und Django (Django Software Foundation n.d.) und/oder QT (The Qt Company n.d.) für das Interface stehen Alternativen auf Basis von Python zur Verfügung. Außerdem beinhalten viele Data Mining Programme wie KNIME, Matlab oder RapidMiner Erweiterungen für Text Mining. Allerdings sind diese Alternativen deutlich komplizierter in der Anwendung und weit weniger unter Linguist_innen verbreitet.

Drittens ist R selbst eine freie, plattformunabhängige Programmiersprache. Dies bedeutet, dass keine Lizenzen gekauft werden müssen. Außerdem kann HanConc sowohl offline als auch

als Desktopversion auf Windows, Linux und Apple Betriebssystemen ausgeführt werden. Die Ausführung auf einem Server kann mit gängigen Linux Distributionen erfolgen. Es ist somit die Möglichkeit eines plattformunabhängigen Einsatzes gegeben.

5.1 Rahmenbedingungen und Projektumgebung für die Erstellung einer Korpussoftware für das Fachsprachenzentrum der Leibniz Universität Hannover (LUH)

In diesem und den folgenden Kapiteln sollen die Konzeption und der Entwicklungsstand von HanConc bis zum Zeitpunkt Juli 2016 dargestellt werden.

HanConc wurde im Jahr 2015 am Fachsprachenzentrum der LUH von Sigrun Schroth-Wiechert und dem Autor dieser Arbeit entwickelt. Finanziert wurde das Projekt für ein Jahr mit einer 50% E13 Stelle über das Programm Konzepte und Ideen für Qualität im Studium (KIQS). Der bewilligte Antrag umfasste drei zu erstellende Korpora (Englisch, Deutsch und Russisch), welche nach Fakultäten geordnete Dissertationen enthalten sollten. Für die ersten beiden Sprachen wurden die Dissertationen durch die Technische Informationsbibliothek und Universitätsbibliothek (TIB/UB) bereitgestellt. Die russischen Texte sollten durch die Polytechnische Staatliche Universität St. Petersburg (SPbSPU)[1] beigefügt werden. Dies ist aufgrund von organisatorischen Schwierigkeiten nicht passiert. Dissertationen wurden als angemessene Textgrundlage ausgewählt, da die Antragsstellerin, Frau Schroth-Wiechert, in ihrer damaligen Tätigkeit bereits Studierende bei ihren Master- und Doktorarbeiten betreut hatte.

Die hier beschriebene Version von HanConc entspricht dem Ergebnis zum Ende des Projektes. Allerdings wurden seitdem einige Veränderungen in den Programmroutinen vorgenommen, um die Perfomance und Stabilität zu verbessern. In der Zwischenzeit wurde eine vereinfachte Version auf der Seite des Fachsprachenzentrums veröffentlicht (Stand 2018).

Die Grundidee dieser Arbeit ist es, Studierenden und ihren Schreibberater_innen relevante wissenschaftliche Texte in ausgewerteter Form für Schreibberatungen zur Verfügung zu stellen. Dissertationen wurden als relevante Texte für Studierende, die ihren Bachelor- oder Masterabschluss anstreben, identifiziert, da es sich hierbei um die Abschlussarbeiten des nächsthöheren Abschlusses handelt (siehe Kapitel 4.1). Diese Dissertationen, auch wenn sie auf dem Campus über die Bibliothek kostenlos und in Gänze abgerufen werden können, unterliegen dennoch dem Urheberrecht. Die Europäische Union (EU) schreibt in Absatz 5 ihrer Richtlinie 2019/790: „In den Bereichen Forschung, Innovation, Bildung [...] ermöglicht die Digitaltechnik neue Nutzungen, die von den geltenden Unionsvorschriften über Ausnahmen und Beschränkungen nicht eindeutig abgedeckt sind." Es ergeben sich hieraus zwei Herausforderungen:

1. Da schon die EU keine Eindeutigkeit der Rechtsgrundlage für Text- und Datamining zu Unterrichtszwecken feststellen kann, ist es dem Autor dieses Textes auch nicht möglich.

[1] Die LUH pflegt eine lange Partnerschaft mit der SPbSPU.

Noch viel weniger kann die Rechtmäßigkeit einer Übergabe der bearbeiteten Texte an Studierende beantwortet werden.

2. Werden Onlineservices zum Zwecke der Textanalyse verwendet, so ist zu beachten, dass bei nicht-europäischen Anbietern, der europäische Rechtsraum verlassen wird.

Wegen der rechtlichen Unsicherheiten wird somit besonders auf Faktoren hingewiesen, die den Einsatz im nicht-europäischen Ausland oder auf lokalen Computern erfordern. Zusätzlich müssen bei der Betrachtung von bestehenden Softwarelösungen und der Konzeption von HanConc die Erkenntnisse aus den Kapiteln 1 bis 4 berücksichtigt werden.

5.2 Analyse von bereits existierender Korpussoftware mit Hinblick auf den Einsatz in Schreibberatungen

Es gibt bereits eine Vielzahl an Software für Schreibzentren und Korpuslinguist_innen. In diesem Kapitel werden einzelne alternative Werkzeuge vorgestellt. Aufgrund der großen Anzahl an Funktionen wird jedoch auf eine umfassende und detaillierte Beschreibung verzichtet. Daher werden Funktionen, die einzigartig für die jeweiligen Programme sind, fokussiert betrachtet. Außerdem wird auf technische Voraussetzungen eingegangen.

Bei der Betrachtung von bestehenden Lösungen werden nur solche korpuslinguistischen Werkzeuge vorgestellt, die es den Nutzer_innen erlauben, eigene Texte zu analysieren. Aus diesem Grund werden etwa die Analysemöglichkeiten des Corpus of Contemporary American English (COCA) oder des British National Corpus (BNC) von der Betrachtung ausgeschlossen. Gleiches gilt für Werkzeuge wie CasualConc[2], die nur auf einem Betriebssystem laufen oder nicht über eine Dokumentation oder Begleitpublikation verfügen wie OpenConc[3]. Ebenso wird auf linguistische Programme verzichtet, die zwar Texte analysieren können, jedoch über keine Volltextsuche verfügen. Als Beispiel hierfür sei etwa Coh-Metrix genannt (McNamara et al. 2014). Pakete wie tm (Feinerer, Hornik & Meyer 2008) und corpustools (Welbers & van Atteveldt 2020) für R oder NLTK (Bird et al. 2009) für Python, die kein eigenes Frontend haben, werden als Werkzeuge für die Funktionen von HanConc an den entsprechenden Stellen diskutiert. Auf eine Diskussion von Software, die zu anderen Zwecken programmiert wurde, aber für Volltextsuchen eingesetzt werden könnte, wird ebenso verzichtet[4].

5.2.1 Open Corpus Workbench (CWB)

Die Open Corpus Workbench (CWB) wurde seit den frühen 1990er Jahren am Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart entwickelt (Christ 1994, Christ, Schulze,

[2] https://sites.google.com/site/casualconc (Stand: 10. März 2020)

[3] https://github.com/muranava/openconc (Stand: 10. März 2020)

[4] Zum Beispiel sei auf Volltextsuchengines wie Lucene und Solr (McCandless, Hatcher & Gospodnetić 2010) oder Log Analyse Software wie Splunk oder LogRhythm verwiesen.

Hofmann & Koenig 1999, Evert & Hardie 2011). Die aktuelle Version 3.5 liegt als quelloffener Source Code, als kompilierter Download oder als fertige virtuelle Maschine vor. In jedem Fall kann die CWB sowohl über die Kommandozeile als auch über ein Webinterface (Corpus Query Processor Web; CQPWeb) aufgerufen werden. Das Webinterface ist in diesem Fall in PHP und die eigentlichen linguistischen Funktionen sind in Perl geschrieben. Zusätzlich stehen noch APIs zu Python und R zur Verfügung[5]. In dieser Arbeit sollen die Möglichkeiten des Webinterfaces diskutiert werden, da dieses eher dem Anwendungsfall einer universitären Schreibberatung entspricht als eine Abfrage mittels Kommandozeile.

Die Texte für die Korpora in der CWB müssen, bevor sie durchsucht werden können, erst noch in ein eigenes Format übertragen werden. Annotationen können vor der Formatierung als Extensible Markup Language (XML) zum Text hinzugefügt werden. Die annotierten Texte werden dann in ein binäres, komprimiertes und indiziertes Format übertragen. Hierdurch wird der notwendige Festplattenspeicher und die Abfragezeit reduziert. Auf die Annotationen kann später über das Interface zugegriffen werden. Außerdem liegen die ursprünglichen Texte in der CWB im Binärformat vor, wodurch eine Manipulation zu einem späteren Zeitpunkt nicht möglich ist.

Suchabfragen im Webinterface werden in einem Suchfeld formuliert. Hierfür steht eine Abfragesprache zur Verfügung, die sich an Reguläre Ausdrücke/Regular Expressions (RegEx) anlehnt. Durch RegEx können Suchergebnisse auf mehrere Schreibweisen eines Wortes, etwa „color" und „colour", oder unterschiedliche grammatikalische Wortendungen, etwa „Auto" und „Autos", erweitert werden. Zusätzlich zu RegEx auf Zeichenebene kann auf die Annotationsebene zugegriffen werden und so etwa die Suchergebnisse auf gewisse Wortarten reduziert werden. Durch die Kombination an Suchparametern können die Ergebnisse entsprechend präzisiert werden.

Die Ergebnisse einer Suche werden bei Standardeinstellungen als Key Words in Context (KWIC) dargestellt. Nachträglich können die Suchergebnisse noch weiterverarbeitet werden. Neben der Organisation der KWIC durch weitere Einschränkungen oder Sortierungen gibt es die Option, die Ergebnisse auch quantitativ auszuwerten. So kann etwa die Verteilung der einzelnen Suchparameter, die Kollokationen oder die Dispersion dargestellt werden. Außerdem besteht die Möglichkeit, die Ergebnisse herunterzuladen.

Die CWB eignet sich jedoch aus mehreren Gründen nicht für den Einsatz in einer Schreibberatung:

- Die grundsätzliche Architektur ist auf den Einsatz auf einem Server ausgerichtet. Zur Installation werden diverse Linux Distributionen oder aber eine virtuelle Maschine mit Linux vorgeschlagen. Dieses Setup ist bedeutend komplexer als etwa bei AntConc, bei dem es ausreicht eine ausführbare Datei herunterzuladen.

- Suchanfragen werden über eine eigene Abfragesprache in Kombination mit RegEx ausgedrückt. Damit wird zusätzlich zu der linguistischen noch eine weitere Komplexität hinzugefügt.

[5] http://cwb.sourceforge.net/files/CQPwebAdminManual.pdf (Stand: 30. Mai 2021)

- Der Fokus liegt auf der Genauigkeit der Suchanfragen und weniger auf der Auswertung der Suchergebnisse. Im Vergleich zu anderen Werkzeugen ist die Anzahl an unterschiedlichen vor allem quantitativen Auswertungen eingeschränkt. Da vor allem die Auswertungen dabei helfen sollen, zu verstehen, wie und in welchen Kontexten bestimmte Wörter oder Konstruktionen auftauchen, verlagert die CWB diese Aufgabe von automatisierten Auswertungen zu möglichst präzisen Suchanfragen.

Der Fokus der CWB liegt auf der schnellen und komplexen Suche in großen Korpora. Da in diesem Projekt eher kleinere und auf die Bedürfnisse der Studierenden angepasste Korpora verwendet, diese möglichst lokal durchsucht werden sollen und es vor allem auf die Auswertung ankommt, erscheint die CWB eher ungeeignet.

5.2.2 WordSmith Tools (WST)

WordSmith Tools (WST) ist ein kommerzielles Werkzeug für Korpuslinguist_innen. Es wird seit 1996 von der Firma Lexical Analysis Software und der Oxford University Press herausgegeben. Seit 2019 steht Version 7 zur Verfügung (Scott 1998/2019).

WST hat mit „Concord", „Wordlist" und „Keywords" drei Hauptfunktionen. Diese werden durch „Utilities" ergänzt. Textaufbereitungen können mit dem „Text Converter" Werkzeug vorgenommen werden, das zu den „Utilities" gehört. Die Beschreibung der einzelnen Funktionen erfolgt auf Basis des Programms und des Handbuchs.

Die „Concord" Funktion beschreibt die KWIC . Neben den eigentlichen Suchergebnissen mit ihrem näheren Kontext werden auch die Position des Suchergebnisses innerhalb des jeweiligen Textes angegeben. Außerdem wird die gesamte Ergebnisliste in Bezug auf Kollokationen und N-Grams ausgewertet. Über die „Concord" Funktion kann auch auf die Dateiliste und die gesamten Texte im Rohformat zugegriffen werden.

Zusätzlich wird eine Kennzahl namens „Dispersion" berechnet. Hierzu wird der Text in acht gleichmäßig große Stücke unterteilt. Basierend auf Oakes (1998) wird „Dispersion" definiert als:

$$\text{Dispersion} = 1 - \frac{\frac{\sigma}{\mu}}{\sqrt{n}}, \tag{5.1}$$

wobei σ und μ die Standardabweichung und das arithmetische Mittel über alle Teilstücke n darstellen. Mit Hilfe der Dispersion lässt sich abschätzen, ob sich die Fundstellen des Suchwortes in bestimmten Teilen eines Textes häufen. Zusätzlich zur Dispersion wird auch die Verteilung innerhalb des Textes als Plot dargestellt. Es ist an dieser Stelle anzumerken, dass die Dispersion auf der Grundannahme basiert, dass die Treffer innerhalb der Teilstücke normalverteilt sind. Kapitel 5.3.8.6 und Gärtner (2014) deuten darauf hin, dass andere statistische Verteilungen wahrscheinlicher sind und daher die Grundannahme der Dispersion verletzt wird.

Mit der „Wordlist" werden alle Texte eines Korpus' ausgezählt und in eine Liste übertragen. Hierbei wird für jedes unterschiedliche Wort ein eigener Eintrag hinzugefügt und die Anzahl an

Vorkommen ergänzt. Zusätzlich werden noch einige Statistiken berechnet. Diese Statistiken haben als Referenz entweder die einzelnen Wörter oder die Texte. Für die einzelnen Wörter wird neben der Frequenz auch die oben beschriebene Dispersion berechnet. Außerdem wird angegeben, in wie vielen Texten das jeweilige Wort vorkommt. Alternativ wird für die einzelnen Texte das Verhältnis von individuellen Wörtern (Types) zur Anzahl an Vorkommen pro Type (Token) berechnet. Hierbei werden neben einer Type-Token Ratio (TTR) für den gesamten Text auch eine TTR pro 1.000 Wörter ermittelt. Es sei an dieser Stelle auf McCarthy & Jarvis (2010) verwiesen, die einen Zusammenhang von Korpusgröße und TTR nachweisen (siehe auch Kapitel 4.4.4). Zusätzlich erhebt die „Wordlist" Funktion noch diverse Statistiken zu den Charakteristika der enthaltenen Worte auf Zeichenebene wie etwa die mittlere Anzahl an Zeichen pro Wort oder die Anzahl an Worten mit einer bestimmten Zeichenlänge.

Die „WS ConcGram" Funktion erweitert die Definition von N-Grams. Während es sich bei N-Grams um direkt nebeneinander stehende Wörter handelt, zeigt „WS ConcGram" auch Wortkombinationen, die durch eingeschobene Wörter getrennt oder in ihrer Reihenfolge geändert worden sind. Als linguistische Grundlage bezieht sich WST auf einen Text von Cheng, Greaves & Warren (2006). Die von „WS ConcGram" erzeugte Liste an ConcGrams kann nach dem Specific Mutual Information (MI) Score, dem MI3 Score, Log Likelihood, T-Score, Z-Score und Dice Coefficient gefiltert werden. Referenzwerte werden nicht angegeben.

Zusätzliche Funktionen, welche für die eigentliche Korpusanalyse oder eine Schreibberatung nicht notwendig sind, aber dennoch auf der obersten Menüebene aufgeführt sind, werden in der folgenden Auflistung zusammengefasst:

- **Aligner:** Diese Funktion erlaubt es, zwei Texte zeilenweise gegenüberzustellen. Dies kann etwa für Übersetzungen oder unterschiedliche Varianten des gleichen Textes angewendet werden.

- **Character Profiler:** Mit dem Character Profiler kann die Zeichenverteilung innerhalb des Korpus' analysiert werden. Hierbei werden alle Zeichen ausgezählt und aufbereitet. Neben den absoluten und relativen Frequenzen wird außerdem angezeigt, wie häufig die einzelnen Zeichen an den unterschiedlichen Stellen eines Wortes vorkommen.

- **CharGrams:** Ähnlich wie bei N-Grams werden bei den CharGrams Kombinationen analysiert. Im Gegensatz zu den N-Grams passiert dies bei den CharGrams nicht auf Wort- sondern auf Zeichenebene. Je nach Einstellung werden etwa Kombinationen aus drei Buchstaben ausgezählt und als Liste präsentiert. Zusätzlich zu den absoluten und relativen Frequenzen wird noch dargestellt, ob sich die Kombination am Anfang, in der Mitte oder am Ende des Wortes befindet.

- **Corpus Checker:** Der Corpus Checker erlaubt es, die Rohdateien der Korpora auf fehlerhafte Dateien zu überprüfen und aus einem Gesamtkorpus ein Subkorpus zu extrahieren.

- **File Utilities:** Diese Funktion vereinfacht das Arbeiten auf Dateiebene. Es können Dateien verschoben, zusammengefasst oder nach bestimmten Kriterien selektiert werden.

- **File Viewer:** Hierbei handelt es sich um einen Hex-Editor, mit dem einzelne Texte auf Byte Ebene geöffnet werden können. Dies kann vor allem sinnvoll sein, um unlesbare Sonderzeichen in Texten zu identifizieren.

- **Minimal Pairs:** Diese Funktion findet auf Basis einer Wortliste Wörter, die sich nur in einem Buchstaben unterscheiden.

- **Text Converter:** WST kann zwar auch XML und HTML Dateien lesen, jedoch sind die Funktionen auf TXT Dateien ausgelegt. Da XML und HTML semistrukturiert sind, muss für eine Korpusanalyse der Text entsprechend gereinigt werden, sofern sie nicht auf die strukturierenden Elemente abzielt. Mit Hilfe der Text Converter Funktion können Dateien in Unicode Textdateien umgewandelt werden[6].

Version 7 von WST wird laut Handbuch auf Windows Rechnern von Windows XP bis Windows 10 unterstützt. Die tatsächlichen Hardwareanforderungen hängen von der Größe des zu untersuchenden Korpus' ab. Als Oberfläche werden mehrere Programmfenster angeboten. Eine Weboberfläche gibt es jedoch nicht.

WST ist ein verbreitetes Werkzeug für Sprachwissenschaftler_innen. Google Scholar findet allein 10.400 wissenschaftliche Beiträge, in denen WST namentlich erwähnt wird[7]. Auch wenn dieses Tool breite Anwendung findet, so ist es dennoch ein Werkzeug für Fachanwender_innen. Weder die statistische noch die linguistische Komplexität wird aufbereitet und vereinfacht sondern als möglichst rohe Zahl präsentiert. Die Aufbereitung der Texte ist nicht Teil von WST. Im Hinblick auf das Ziel dieser Arbeit, komplexe linguistische und statistische Verfahren für Studierende und Schreibberatende zum Self-Service zur Verfügung zu stellen, erscheint WST in vielen Dimensionen zu komplex in seiner Darstellung und Semantik. In Bezug auf andere Dimensionen, vor allem fortgeschrittenere Algorithmen, erweist sich WST als unterkomplex. Außerdem sorgt die lokale Oberfläche dafür, dass jede Anwender_in von WST kompletten Zugriff auf alle Daten erhält, selbst für die Installation und Wartung zuständig ist und die maximale Korpusgröße durch die Hardware der Studierenden limitiert ist.

5.2.3 AntConc

AntConc gehört zu einer Vielzahl von Software, die von Laurence Anthony entwickelt wurde. Im Gegensatz zu anderen Tools wie WordSmith Tools ist AntConc kostenlos und kann von Laurence Anthonys privater Internetseite heruntergeladen werden[8].

AntConc stand im März 2020 in der Version 3.5.8 für Windows, Mac und Linux zur Verfügung (Anthony 2019). Die Software muss nicht installiert werden, sondern kann direkt aus der EXE[9] Datei heraus gestartet werden. Ein Webinterface ist nicht vorhanden.

[6] Dies gilt nur für XML und HTML. JSON, DOC(X), PPT(X) oder PDF können nicht eingelesen werden.

[7] Diese Zahl wurde mittels einer Suche nach der exakten Wortkombination „WordSmith Tools" ermittelt (Stand: 05. März 2020)

[8] https://www.laurenceanthony.net/software (Stand: 05. März 2020)

[9] Für Mac und Linux ergeben sich Abweichungen.

Statt auf ein Hauptfenster und mehrere Unterfenster für die einzelnen Funktionen setzt AntConc auf einen Hauptbildschirm. Auf der linken Seite werden die geöffneten und durchsuchbaren Dateien aufgelistet und auf der rechten Seite die Ergebnisse der einzelnen Funktionen:

- **KWIC**: Neben einer reinen KWIC Funktion werden auch die Trefferstellen als Plot dargestellt. Die jeweiligen Rohtexte zu den Plots können im „File View" Reiter gelesen werden. Die „Cluster" Funktion zeigt N-Grams und zusätzlich können noch Kollokationen und die Wortliste angezeigt werden.

  Die KWIC Funktion erlaubt Suchen nach einzelnen oder mehreren Wörtern, Case Sensitive Suchen und Suchen mit regulären Ausdrücken. Ergebnisse können alphabetisch nach den Wörtern rechts und links vom Suchwort sortiert werden. Die KWIC erstrecken sich auf eine bestimmte Anzahl an Wörtern rechts und links vom Suchwort und sind unabhängig von Satzgrenzen. Zusätzlich wird bei jedem Ergebnis der Dateiname der entsprechenden Textdatei genannt.

- **Concordance Plots**: Hiermit kann die Position der Suchtreffer in den Texten dargestellt werden. Dabei wird jede Datei in einer Zeile angezeigt.

- **Cluster**: Diese Funktion findet N-Grams entsprechend der Vorgaben und sortiert sie nach ihrer Frequenz. Außerdem wird ermittelt, in wie vielen Texten die N-Grams vorkommen.

- **Collocations**: Kollokationen können mit der entsprechenden Funktion ermittelt werden. Hierbei wird die Frequenz für die rechte und linke Seite angezeigt und eine Statistik berechnet. Dabei kann zwischen einem T-Score, Log-Likelihood, dem MI Score oder einer Kombination aus letzteren gewählt werden.

- **Word List**: In dieser Liste wird das gesamte Korpus ausgezählt und die einzelnen Wörter geordnet nach ihrer Frequenz angezeigt. Um zu überprüfen, ob die Frequenz eines Suchbegriffs im Vergleich zu einem Referenzkorpus statistisch relevant häufiger oder seltener vorkommt, kann die „Keyword List" verwendet werden. Hierfür werden Vier-Feld Kreuztabellen, χ^2 und Log-Likelihood Signifikanztests genutzt.

Im Gegensatz zu anderen linguistischen Softwarepaketen wird die Gesamtheit aller Funktionen über mehrere Programme verteilt. Die nachfolgende Liste zeigt die für eine Schreibberatung möglicherweise relevanten Programme. Diese sind sortiert als erweiterte Funktionen von AntConc und Werkzeuge zur Textaufbereitung:

- **AntGram:** Generiert Wortcluster und N-Gram Listen ähnlich der gleichnamigen Funktion in AntConc

- **AntPConc:** Hierbei handelt es sich um ein Konkordanztool für parallele Korpora.

- **AntMover:** Analysiert die pragmatische Textstruktur. Es ist zu bedenken, dass diese Funktion erst manuell trainiert werden muss.

- **ProtAnt:** Diese Funktion ermittelt die Prototypikalität eines Textes. Allerdings werden hierbei zwei Korpora verglichen und die Prototypikalität ergibt sich aus dem Vergleich der Texte innerhalb eines Korpus im Gegensatz zum Referenzkorpus.
- **VariAnt:** Findet mehrere Schreibweisen eines Wortes
- **AntWordProfiler:** Berechnet oberflächliche Komplexitätskennzahlen
- **TagAnt:** Ist ein Interface zum TreeTagger (Schmid 1999)
- **AntCLAWSGUI:** Hierbei handelt es sich um ein Interface zum CLAWS Tagger.
- **EncodeAnt:** Ändert die Zeichencodierung von Textdateien
- **SarAnt:** Ermöglicht das Suchen und Ersetzen von Zeichen in mehreren Texten
- **AntFileConverter:** Konvertiert PDF und Word Dokumente in TXT Dateien
- **AntFileSplitter:** Zerteilt Texte nach einer bestimmten Wortanzahl in mehrere Dateien
- **AntCorGen:** Generiert englischsprachige Korpora aus PLOS ONE Artikeln

Ebenso wie WST richtet sich AntConc an Sprachwissenschaftler_innen. Die Funktionen und Ergebnisse werden mit Fachbegriffen gekennzeichnet, Kennzahlen werden ohne Referenzwert angegeben und auch nicht weiter aufbereitet. Außerdem fehlen auch hier Verfahren, die über das oberflächliche Auszählen von Wörtern hinausgehen oder statistisch motiviert sind. Mit Hinblick auf die Erkenntnisse aus den bisherigen Kapiteln deutet sich auch hier an, dass AntConc kaum für den Einsatz in Schreibberatungen geeignet ist.

5.2.4 Corpkit

Im Gegensatz zu WST und AntConc ist Corpkit nicht nur kostenlos sondern auch Open Source Software (McDonald 2015). Der Quellcode liegt in einem Git Repository[10] vor. Da es sich um ein Python Paket handelt, kann der Quellcode ebenfalls nach der Installation von der Festplatte gelesen werden.

Corpkit bietet mehrere Möglichkeiten der Interaktion: Es kann entweder über den Python Interpreter, über eine Python API oder über eine grafische Benutzeroberfläche auf Tk[11] Basis bedient werden. Corpkit ist damit unabhängig vom Betriebssystem und kann gegebenenfalls auch zu einer Webanwendung umfunktioniert werden.

Im Gegensatz zu den vorherigen Tools bietet Corpkit eine Reihe an Textaufbereitungswerkzeugen. Hierbei werden die Funktionen des Stanford CoreNLP Pakets (Manning, Surdeanu, Bauer, Finkel, Bethard & McClosky 2014) verwendet. Es wird daher zusätzlich zu Python noch eine Java Installation benötigt. Neben der notwendigen Tokenisierung können auch die Sätze

[10] https://github.com/interrogator/corpkit (Stand: 10. März 2020)
[11] https://docs.python.org/3/library/tk.html (Stand: 05. März 2020)

auf jeweils einzelne Zeilen verteilt, die einzelnen Wörter mit PoS Tags versehen, benamte Entitäten ermittelt und ein syntaktischer Parser auf die Sätze angewandt werden. Bei Dialogen können die individuellen Sprecher_innen mittels ihrer ID am Zeilenanfang identifiziert und einzeln analysiert werden.

Suchen werden in Corpkit als „Interrogations" also Befragungen bezeichnet. Hierbei können direkte Suchbegriffe aber auch RegEx verwendet werden. Bei RegEx kommt die in Python integrierte Engine zum Einsatz. Neben Wörtern können auch Lemmata, PoS Tags, Named Entity Recognition (NER) Tags und Elemente aus dem syntaktischen Parser als Anker für die Suchen dienen. Außerdem können bestimmte Elemente aus der Suche ausgeschlossen werden.

Corpkit bietet neben den KWIC auch noch andere Suchergebnistypen. Die Ergebnisse der Suche können als ausgezählte N-Grams und Kollokationen erzeugt und pro Abschnitt ausgewertet werden. Die Abschnitte werden bei Einlesen der Texte anhand von Leerzeilen in der Textdatei erzeugt. Statistische Auswertungen finden nicht statt.

Alle Ergebnisse können mit der Python Bibliothek Matplotlib visualisiert oder als Pandas Dataframe, Python Objekt oder Python Dictionary exportiert und weiter verarbeitet werden.

Die Hauptzielgruppe von Corpkit besteht aus linguistischen Expert_innen. Daher wird für die Suchanfragen und Ergebnistypen ein entsprechendes Fachwissen vorausgesetzt. Die technische Grundidee von Corpkit entspricht der von HanConc. Beide Werkzeuge sind als Open Source Software auf einfache Manipulation und Offenheit ausgelegt. Corpkit könnte mit etwas Arbeit in ein Webframework integriert werden. Der größte Unterschied ergibt sich jedoch aus dem linguistischen Fokus beider Ansätze. Während sich die Suchkomplexität bei HanConc aus der Kombination einzelner Elemente ergibt, so setzt Corpkit auf Textannotationen und RegEx. HanConc kann auf Satzebene nach Wörtern, eingeschränkt nach Lemmata und nach Wortarten suchen, wohingegen Corpkit auch syntaktische Parser und NER unterstützt. Allerdings beschränken sich die Ergebnisse auf Auszählungen, einfache N-Grams, Kollokationen und aufwendige grafische Darstellungen.

Die einzige Dokumentation von Corpkit befindet sich eingeschränkt auf GitHub und im Quellcode. Daher können leider nicht alle Funktionen genauer betrachtet und beschrieben werden. Der Quellcode ist außerdem nur in Teilen lauffähig. Für einen Einsatz in einer Schreibberatung bräuchte es daher eine technische Überarbeitung, ein Aufbereiten der Ergebnisse und gegebenenfalls eines neues Frontend. Zusätzlich ergibt sich noch die Schwierigkeit, dass Corpkit in Python geschrieben ist und daher vermutlich eine höhere Einstiegshürde für Schreibberater_innen besteht.

5.2.5 WordStatix

WordStatix wurde von Massimo Nardello in Free Pascal entwickelt und steht kostenlos und als Open Source Software in der aktuellen Version 1.9.0 für Windows, Linux und Mac zur Verfügung[12] (Nardello 2016). In dieser Arbeit wird die Windows Version betrachtet.

[12] https://sites.google.com/site/wordstatix/home (Stand: 10. März 2020)

Nach der Installation bietet WordStatix ein Programmfenster ähnlich denen von WST und AntConc an. Auch hier sind die einzelnen Analysemöglichkeiten in Reitern organisiert. Neben dem Rohtext stehen auch eine KWIC Funktion, Statistiken und Diagramme zur Verfügung. Im Gegensatz zu WST und AntConc kann nur eine Datei gleichzeitig geöffnet werden. Allerdings können neben TXT Dateien auch DOC und ODT Dateien geöffnet werden.

Nach dem Öffnen des zu analysierenden Korpus muss zuerst die Konkordanz erstellt werden. Die Performanz dieses Arbeitsschrittes ist jedoch im Vergleich zu den anderen Werkzeugen deutlich schlechter. Während WST und AntConc Sekunden brauchen, um das FElt Korpus mit etwa 1.800.000 Wörtern zu durchsuchen, braucht WordStatix allein zur Erstellung der Konkordanz über 90 Minuten. Leider steht WordStatix auch nur als 32 Bit Programm zur Verfügung, sodass auch nur ein Bruchteil der Hardware ausgenutzt wird[13].

Im Gegensatz zu den anderen Korpuswerkzeugen wird bei WordStatix zuerst eine Wortliste erstellt, welche als Startpunkt für die KWIC dient. Anstatt direkt in den Text zu springen, findet die Suche das jeweilige Wort in der Wortliste. Wird ein Wort markiert, so werden auf dieser Basis die KWIC angezeigt. Dieses Vorgehen wird Inverted Index genannt (Cutting & Pedersen 1989) und bietet den Vorteil, dass bei einer Suche nicht das gesamte Korpus betrachtet werden muss, sondern direkt an die Zielstellen gesprungen werden kann[14]. Der Preis für den Inverted Index ist die lange Aufbereitungszeit zu Beginn des Prozesses.

Neben der Suche nach einem oder mehreren Wörtern können auch Präfixe und Suffixe verwendet werden, um alle relevanten Suchbegriffe in der Wortliste zu selektieren und damit in die KWIC aufzunehmen. Die zusätzlichen Statistik- und Diagrammfunktionen beziehen sich auf die absolute Anzahl an Wörtern pro Abschnitt, wobei diese aus dem Rohtext extrahiert werden. Die Statistikfunktion ermittelt, wie häufig die Suchbegriffe in den einzelnen Abschnitten vorkommen. Hierbei werden absolute Zahlen angezeigt. Die Diagrammfunktion überführt diese Zahlen in Balkendiagramme. Funktionen zur Textaufbereitung stehen nicht zur Verfügung.

Für die Arbeit in einer Schreibberatung erscheint WordStatix aus den gleichen Gründen wie WST und AntConc eher ungeeignet. Erschwerend kommt hinzu, dass im Vergleich zu den oben besprochenen Werkzeugen weniger Funktionen zur Verfügung stehen und gleichzeitig die initiale Berechnung der Konkordanz beziehungsweise Wortliste vergleichsweise lange dauert[15].

5.2.6 ShinyConc

Was die Architektur der Anwendung angeht, steht ShinyConc HanConc am nächsten. ShinyConc basiert ebenso auf einem R Shiny Frontend mit einem in R programmierten Backend. Der Konkordanzer kann als Quellcode von GitHub geladen[16], manuell mit Korpora und Metadaten bestückt und selbstständig in Betrieb genommen werden. Alternativ kann über eine Onli-

[13] Durch die 32 Bit Architektur wird nur ein einzelner Prozessorkern verwendet.

[14] Bibelkonkordanzen funktionieren ebenfalls nach diesem Prinzip: Für jedes Suchwort sind alle Bibelstellen angegeben, an denen dieses Wort vorkommt. Anstatt die gesamte Bibel nach dem Suchwort zu durchsuchen, wird die Konkordanz nach dem Suchwort durchsucht und mit Hilfe der Angaben direkt an die jeweiligen Stellen in der Bibel gesprungen.

[15] Das Einlesen des FElt Korpus zu Testzwecken hat bei 3,07 GHz pro CPU Kern über 90 Minuten gedauert.

[16] https://github.com/cwolk/ShinyConc (Stand: 10. März 2020)

neanwendung namens „ShinyConc Builder" eine fertige Shiny App inklusive der Textgrundlage generiert und als Download zur Verfügung gestellt werden[17]. Hierfür müssen die Texte und Metadaten auf die angegebene Seite hochgeladen werden. Bei der zweiten Lösung ist allerdings zu bedenken, dass die Datenschutzhinweise auf der Builder Seite fehlen und daher das Hochladen von Daten auf eigenes rechtliches Risiko erfolgt.

ShinyConc benötigt für das initiale Aufsetzen einer Korpusanalyse die Texte und eine Metadatendatei, aus der die Subkorpora gebildet werden. Die zugrunde liegende Datei muss daher mindestens eine Spalte mit den Dateinamen enthalten. Jede weitere Spalte dient zur Eingrenzung auf weitere Merkmale. Im Falle der in dieser Arbeit untersuchten Dissertationen der LUH würde die Tabelle die Dateinamen und die jeweilige Fakultät beinhalten. Da die Unterteilung in einzelne Korpora über die Metadaten erfolgt, können alle Dateien in einem Ordner gespeichert werden.

Leider gibt die online verfügbare Dokumentation[18] wenig Aufschluss über die genaue Funktionsweise von ShinyConc. Der Quellcode zeigt jedoch, dass die Texte als Charaktervektor eingelesen werden. Als einziger Aufbereitungsschritt werden beim Einlesen überzählige Leerzeichen entfernt ($loadcorpus.R$). Die eigentliche Suche erfolgt über das stringr Paket (Wickham 2019). Hierbei kann nach einzelnen Wörtern oder mit Hilfe von RegEx gesucht werden.

Die Ergebnisse einer Suche werden als KWIC dargestellt. Der Kontext ist dabei auf wenige Zeichen rechts und links des Suchwortes beschränkt. Allerdings kann der gesamte Text durch Markieren der Zeile und die Schaltfläche „Show Full Text" angezeigt werden. Neben den KWIC kann auch die Anzahl an Treffern pro Suchwort[19] und eine ausgezählte Wortliste angezeigt werden.

Über die „Compare" Funktion können zwei Subkorpora miteinander verglichen werden. Hierbei wird das gleiche Gesamtkorpus anhand der Metadaten nach Auswahl der Nutzer_in in zwei Subkorpora getrennt. Für beide Korpora werden die KWIC ausgezählt und als absolute und relative Frequenz dargestellt. Zusätzlich wird eine Kennzahl „Keyness" aufgeführt. Aus dem Quellcode ergibt sich, dass es sich hierbei um einen χ^2 Wert für eine Vier-Feld Kreuztabelle der absoluten Zahlen je Korpus handelt. Diese Kennzahl wird jedoch in ShinyConc nicht weiter thematisiert.

Im Vergleich zu den vorherigen Konkordanzprogrammen ist der Funktionsumfang von ShinyConc deutlich limitierter. Ebenso schränkt die Verwendung von stringr als Suchengine die Möglichkeiten einer Erweiterung der Texte um PoS Tags, Lemmata und syntaktische Komponenten deutlich ein, da nur einzelne Charactervektoren durchsucht werden können und zusätzliche Annotationen dann über RegEx gesucht werden müssten. Allerdings ist ShinyConc durch seine Multiplattformarchitektur, den offenen Quellcode und die vergleichsweise einfache Programmierung an die Anforderungen einer Schreibberatung anpassbar.

[17] http://shinyconc.de/builder/ (Stand: 10. März 2020)

[18] http://shinyconc.de/tutorial.pdf (Stand: 10. März 2020)

[19] Wird mit RegEx gesucht, können mehrere Wörter gefunden werden. In diesem Fall wird die Anzahl an Treffern je Suchwort berechnet.

5.2.7 SketchEngine

SketchEngine ist ein kommerzielles Produkt. Im Gegensatz zu etwa WST, welches zwar Geld kostet, jedoch aus einer Forschungseinrichtung heraus entstanden ist und auch für solche Einrichtungen entwickelt wurde, zielt SketchEngine auf kommerzielle Nutzer_innen[20] ab. Da es sich um ein Onlinewerkzeug handelt, müssen die eigenen Texte zur Analyse auf die Server von SketchEngine übertragen werden. Weil der Hauptsitz von SketchEngine in Brno, Tschechische Republik, liegt, muss sich SketchEngine an die europäische Datenschutzgrundverordnung halten und gibt entsprechende Hinweise in der Datenschutzerklärung[21]. In den Nutzungsbedingungen wird allerdings darauf hingewiesen, dass die Nutzer_in die Rechte an den Texten halten muss[22]. Aus diesem Grund wird bei der Analyse und Beschreibung von SketchEngine auf die Verwendung der in dieser Arbeit beschriebenen Korpora verzichtet und stattdessen die in SketchEngine geladenen deTenTen11 und deTenTen13 Korpora genutzt. Bei diesen Korpora handelt es sich um Webkorpora aus den Jahren 2011 und 2013.

Bei der Startfunktion „Word Sketch" werden die Ergebnisse der Suche kategorisiert und mit Beispielen versehen. Anhand der Wörter „deshalb" und „daher" sollen die Funktionen von SketchEngine beschrieben werden. Die Ergebnisse der „Word Sketch" Funktion werden zweigeteilt dargestellt (siehe Abbildung 5.1): Einerseits werden Adjektive angezeigt, die von „deshalb" modifiziert werden, und andererseits werden von „deshalb" modifizierte Verben aufgelistet. Die Sortierung erfolgt anhand des logarithmierten Dice Koeffizienten. Dieser ist definiert als (Rychlỳ 2008):

$$\text{Log Dice} = 14 + log_2 \frac{2 \cdot \|w1, R, w2\|}{\|w1, R, *\| + \|*, R, w2\|} \tag{5.2}$$

Dabei wird davon ausgegangen, dass es sich um ein Triplet bestehend aus einem ersten Wort von Interesse, $w1$, einem grammatikalischen Funktionswort, R, und einem zweiten Wort von Interesse, $w2$ handelt. Der Log Dice Koeffizient ist definiert von minus unendlich bis 14.

Von den einzelnen Wörtern kann in die Konkordanzfunktion mit den entsprechenden Textstellen oder in den Thesaurus gesprungen werden.

Die „Concordance" Funktion beinhaltet sowohl die KWIC als auch diverse andere Funktionen zur weiteren Analyse der Treffer. Es besteht die Möglichkeit, die Ergebnisse zu sortieren, den Kontext auf Sätze oder Wörter zu beziehen oder sich die Frequenzen der umgebenden Wörter und Wortarten anzeigen zu lassen.

Zusätzlich zu den gängigen KWIC Funktionen können Good Dictionary Examples (GDEX) Sätze angezeigt werden (Kilgarriff, Husák, McAdam, Rundell & Rychlý 2008). Die genaue Definition von GDEX ist nicht veröffentlicht. Laut der in der Dokumentation referenzierten Artikel (Kilgarriff et al. 2008) ist ein GDEX zumindest für Englisch nach folgenden Parametern definiert:

[20] Die Webseite von SketchEngine listet unter anderem Cambridge und Oxford University Press und den deutschen Cornelsen Verlag als Kunden auf (Stand: 10. März 2020).

[21] https://www.sketchengine.eu/gdpr-privacy-consent/ (Stand: 10. März 2020)

[22] "Users are responsible for the copyright and other intellectual property issues of content uploaded by themselves." auf https://www.sketchengine.eu/terms-of-use/ (Stand: 10. März 2020)

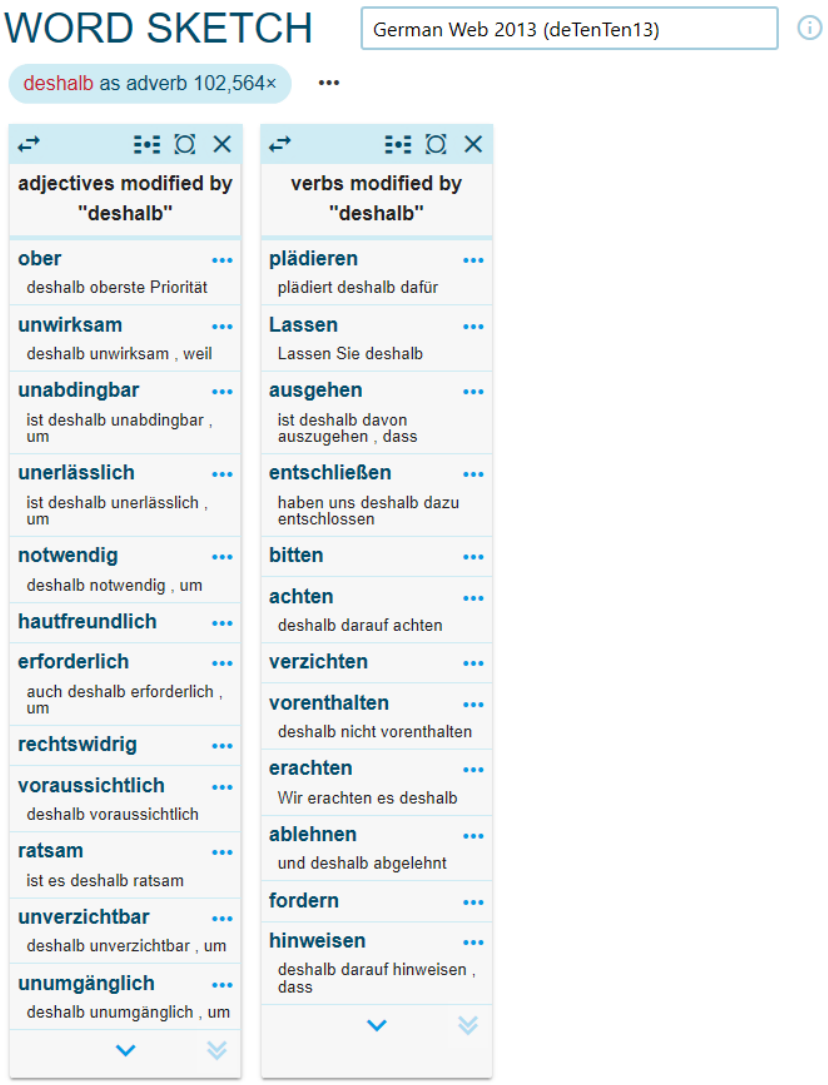


Abbildung 5.1: Word Sketch für das Wort „deshalb” im deTenTen13 Korpus

- Ein Satz bestehend aus 10 bis 25 Wörtern
- Keine bis wenige Pronomina
- Alle Wörter des Satzes gehören zu den 17.000 häufigsten Wörtern
- Ein kompletter Satz, wobei dieser durch einen initialen Großbuchstaben und ein abschließendes Satzzeichen definiert ist
- Beinhaltet Kollokationen mit einer hohen Bindekraft
- Das Suchwort befindet sich in der Mitte oder am Ende des Satzes

Abweichungen von dieser Definition führen nicht zum Ausschluss des Satzes, sondern werden als Strafwerte mit in die Kalkulation gegeben. Die genaue Gewichtung der einzelnen Werte ergibt sich aus einem Experiment der Autor_innen. Hierbei wählten zwei Student_innen subjektiv für gut befundene Bespiele für 1.000 Kollokationen aus. Um zu der Gewichtung und damit zur genauen Kalkulation zu gelangen, wurde versucht, die Auswahl der beiden Student_innen kalkulatorisch zu reproduzieren (Kilgarriff et al. 2008). Die Ergebnisse und die Formel für die Kalkulation werden nicht beschrieben. Auch weicht die Zusammensetzung der Kalkulation in unterschiedlichen Artikeln zum Thema voneinander ab (Kosem, Husak & McCarthy 2011).

„Word Sketch Difference" beschreibt zwei Wörter kontrastiv. Ähnlich wie bei der „Word Sketch" Funktion werden Kollokationen ausgewertet. Bei dieser Funktion werden jedoch die absoluten Werte dargestellt. Die daraus resultierende Tabelle ist in drei Kategorien unterteilt. Die erste Kategorie enthält Kollokationen, die deutlich häufiger mit dem ersten Wort benutzt werden, die dritte Kategorie solche mit stärkerer Tendenz zum zweiten Wort und die mittlere Kategorie listet Wörter auf, die mit beiden Wörtern kollokieren. Abbildung 5.2 zeigt diese Funktion am Beispiel von „daher" und „deshalb" im deutschen deTenTen11 Korpus.

Die „Word List" kann für Wörter, Lemmata oder verschiedene grammatikalische Wortarten angezeigt werden. Zusätzlich kann sie auf Wörter mit spezifischen enthaltenen Zeichenkombinationen eingeschränkt werden. Die Wortliste besteht dann aus der Position, dem jeweiligen Wort oder Lemma und einer Kennzahl. Neben der absoluten Frequenz kann auch die relative Frequenz pro Million Wörter dargestellt werden. Als alternative Referenz zum Gesamtkorpus steht auch die Dokumentenstruktur zur Verfügung, sodass anstatt der absoluten Frequenz im Vergleich zum Gesamtkorpus die Anzahl an Dokumenten, welche das Wort enthalten, angezeigt werden kann. Als dritte Kennzahl wird die Average Reduced Frequency (ARF) angeboten. Hierbei handelt es sich um eine Maßeinheit für die Verteilung innerhalb des Korpus.

Die Beschreibung der ARF ist der entsprechenden Dokumentation auf der Internetseite von SketchEnginge[23] und einem Artikel von Savickỳ & Hlavácová (2002) entnommen. Die ARF für ein Wort wird folgendermaßen bestimmt: Jedes Wort im Korpus erhält eine aufsteigende Positionsnummer. Das Korpus wird in so viele Teile geteilt, wie es Treffer für das zu untersuchende Wort gibt. In diesem Beispiel soll das Korpus 60 Tokens und fünf Treffer enthalten.

[23] https://www.sketchengine.eu/documentation/average-reduced-frequency/ (Stand: 10. März 2020)

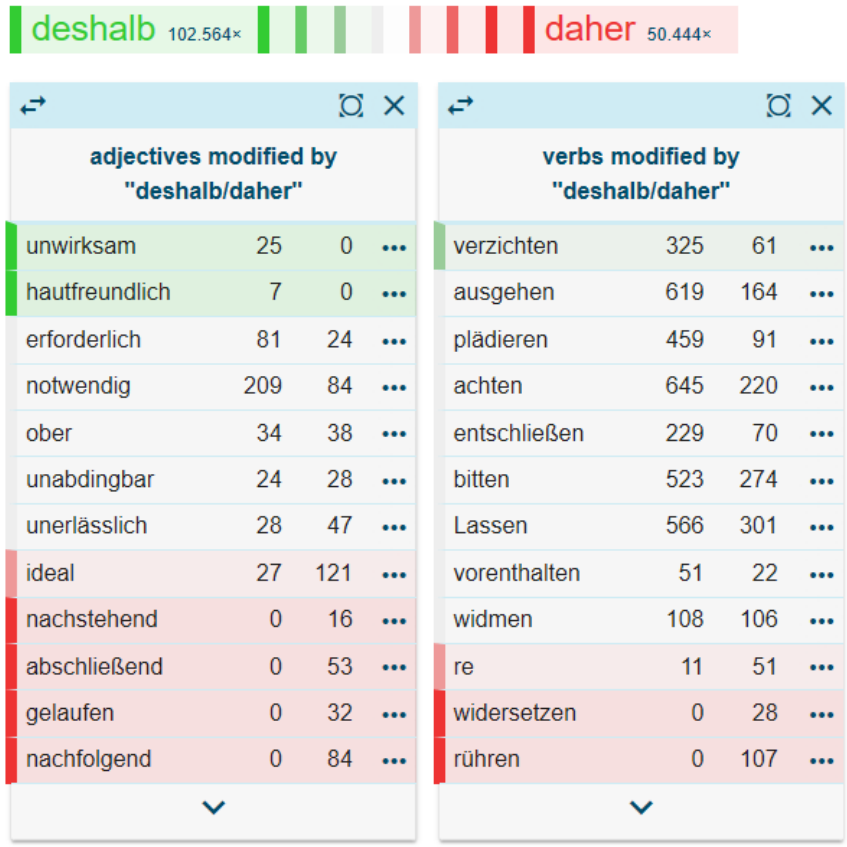


Abbildung 5.2: Word Sketch Difference für „daher” und „deshalb” im deTenTen11 Korpus

Die Treffer sind über das gesamte Korpus verteilt. Nun wird das Korpus in fünf gleichmäßige Teile zu je 12 Tokens geteilt. Die Reduced Frequency ergibt sich aus der Anzahl an Teilen, die mindestens einen Treffer enthalten. Um den Einfluss der Teilgrenzen zu eliminieren, wird die Grenze in mehreren Iterationen soweit nach rechts verschoben, bis sie eine Position vor der ursprünglichen Grenze liegt. Die Reduced Frequencies werden abschließend zur ARF gemittelt. Abbildung 5.3 visualisiert dieses Vorgehen. In ihrer Grundidee ähnelt die ARF damit der Dispersion nach Oakes (1998) in WST. Es werden keine weiteren Informationen im Zusammenhang mit der Wortliste dargestellt.

Der „Thesaurus” zeigt semantisch ähnliche Wörter, die in bestimmten Kontexten synonym verwendet werden können. Um diese zusammenhängenden Wörter zu identifizieren, wird vor allem der umgebende Kontext analysiert. Der Algorithmus und die linguistische Beschreibung hierzu kann in Kilgarriff, Baisa, Bušta, Jakubíček, Kovář, Michelfeit, Rychlỳ & Suchomel (2014), Kilgarriff, Rychly, Smrz & Tugwell (2004), Rychlỳ & Kilgarriff (2007) und Lin (1998) nachgelesen werden. Abbildung 5.4 zeigt die Ergebnisse der Thesaurusfunktion, welche jedoch hier nicht weiter diskutiert werden. Die Ausführungen und Experimente in Rychlỳ & Kilgarriff (2007) deuten darauf hin, dass eine Umsetzung der Thesaurusfunktion in R auf einem Heimcomputer die Laufzeit abhängig von der Korpusgröße um mehrere Tage oder Wochen erhöhen würde.

Ähnlich wie die Wortliste werden die Ergebnisse der „N-Gram” Funktion als Tabelle dargestellt. Als Standard werden Tri-Grams, also Dreiwortkombinationen, erzeugt. Jedoch kann

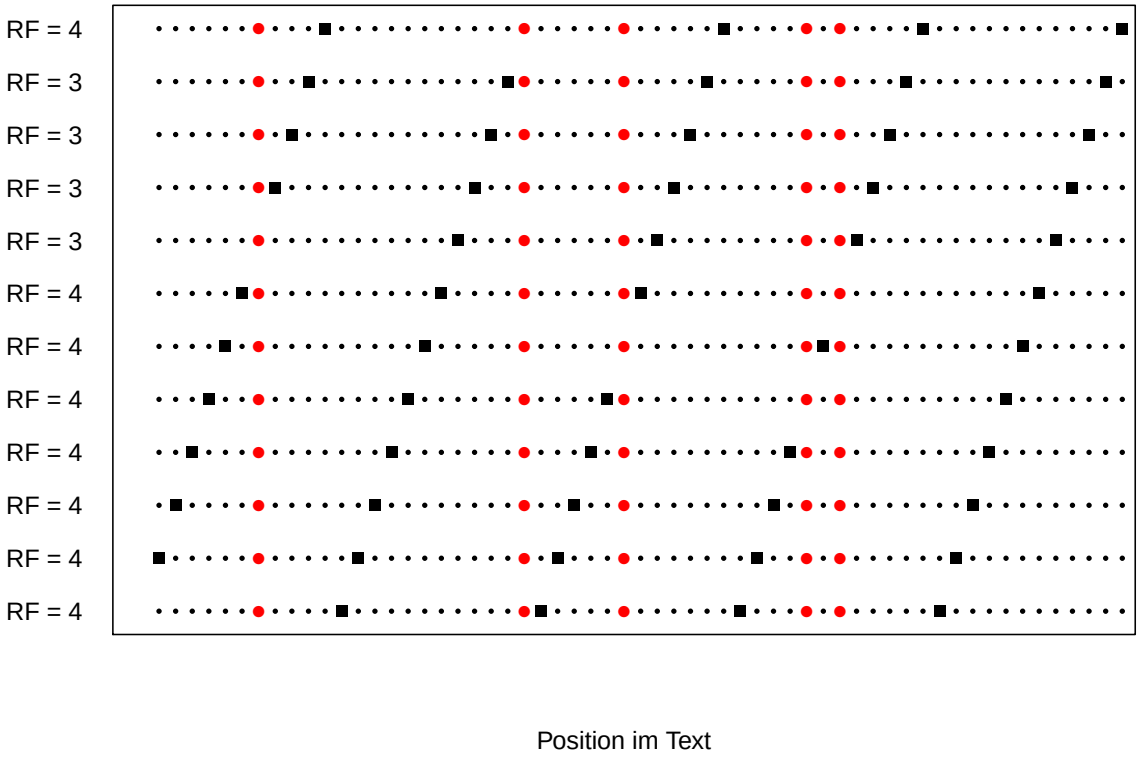
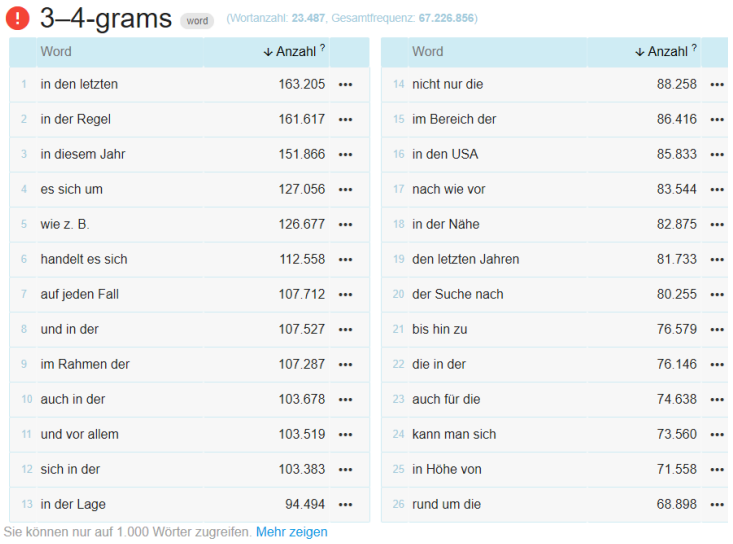


Abbildung 5.3: Average Reduced Frequency (ARF) von $3,\overline{6}$ in einem Beispielkorpus mit 60 Tokens und 5 Treffern; Tokens sind mit schwarzen Punkten, Treffer mit größeren roten Punkten und Grenzen mit schwarzen Quadraten markiert

	Word	Frequenz ?	
1	folglich	249.807	•••
2	daher	50.444	•••
3	überdies	137.235	•••
4	irgend	279.012	•••
5	zweitens	294.816	•••
6	ferner	456.088	•••
7	indes	231.566	•••
8	erstens	314.435	•••
9	wenn	498.395	•••
10	seinerseits	96.430	•••

Abbildung 5.4: Thesauruseinträge für „deshalb” im deTenTen11 Korpus

die Größe der N-Grams von zwei bis zu sechs Elementen variiert werden. Wird eine N-Gram Größe von mindestens drei gewählt, so können kleine N-Grams in größere geschachtelt werden. Abbildung 5.5a zeigt die flache N-Gram Liste für das deTenTen11 Korpus. Die Liste ist absteigend nach Anzahl geordnet. Eine Suche innerhalb der Liste ist nicht möglich. Wird die Option für geschachtelte N-Grams ausgewählt, ändert sich die Darstellung dahingehend, dass unter den größeren N-Grams kleinere N-Grams aufgeführt sind. In dem Beispielauszug in Abbildung 5.5b werden Dreiwortkombinationen in Vierwortkombinationen geschachtelt. Die blauen Zahlen hinter den Tri-Grams zeigen die entsprechenden Frequenzen. In beiden Fällen kann von den N-Grams in die Konkordanz des ausgewählten N-Grams gesprungen werden. Es ist zu bedenken, dass nur die erste Milliarde Tokens berücksichtigt wird.



3–4-grams word (Wortanzahl: 23.487, Gesamtfrequenz: 67.226.856)

	Word	Anzahl
1	in den letzten	163.205
2	in der Regel	161.617
3	in diesem Jahr	151.866
4	es sich um	127.056
5	wie z. B.	126.677
6	handelt es sich	112.558
7	auf jeden Fall	107.712
8	und in der	107.527
9	im Rahmen der	107.287
10	auch in der	103.678
11	und vor allem	103.519
12	sich in der	103.383
13	in der Lage	94.494
14	nicht nur die	88.258
15	im Bereich der	86.416
16	in den USA	85.833
17	nach wie vor	83.544
18	in der Nähe	82.875
19	den letzten Jahren	81.733
20	der Suche nach	80.255
21	bis hin zu	76.579
22	die in der	76.146
23	auch für die	74.638
24	kann man sich	73.560
25	in Höhe von	71.558
26	rund um die	68.898

Sie können nur auf 1.000 Wörter zugreifen. Mehr zeigen

(a) N-Grams bestehend aus drei oder vier Wörtern im deTenTen11 Korpus

27	gibt es auch		68.151
28	handelt es sich um • es sich um 127.056 • handelt es sich 112.558		67.927
29	es in der		67.863
30	im Rahmen des		67.763
31	sich in den		67.183
32	in den letzten Jahren • in den letzten 163.205 • den letzten Jahren 81.733		66.468

(b) Genestete N-Grams bestehend aus drei oder vier Wörtern im deTenTen11 Korpus

Abbildung 5.5: Beispielhafte Ergebnisse der „N-Gram" Funktion bestehend aus drei oder vier Wörtern im deTenTen11 Korpus

Die Funktion „Keywords / Terminology Extraction" steht für deutsche Korpora nicht zur Verfügung. Potentielle Keywords wie sie in Abbildung 5.6 für den Vergleich des BNC zum English Web 2013 (enTenTen13) Korpus aufgeführt werden, sind jene Wörter, die typisch für ersteres Korpus im Kontrast zum zweiten Korpus sind. Entsprechend der Dokumentation von SketchEngine und dem darin aufgeführten Artikel (Kilgarriff 2012) ergeben sich die Keywords aus:

$$\text{Keyness} = \frac{\frac{f_{KorpusA}}{1.000.000} + N}{\frac{f_{KorpusB}}{1.000.000} + N} \tag{5.3}$$

Hier werden zwei relative Frequenzen ins Verhältnis gesetzt. Der Scheitelpunkt des Vergleichs liegt bei eins. Die gesamte Funktion ist definiert von null bis eins für eine häufigere Verwendung

	Word			Word			Word	
1	erm	•••	11	innit	•••	21	aye	•••
2	er	•••	12	Gentleman	•••	22	Swindon	•••
3	Mhm	•••	13	Ended	•••	23	Gloucester	•••
4	cos	•••	14	Ooh	•••	24	Aha	•••
5	LIFESPAN	•••	15	EC	•••	25	Unix	•••
6	Darlington	•••	16	Ref	•••	26	nineteen	•••
7	Cos	•••	17	Aye	•••	27	Gloucestershire	•••
8	Yeah	•••	18	Labour	•••	28	Pounds	•••
9	Middlesbrough	•••	19	Began	•••	29	yeah	•••
10	Mrs	•••	20	Kinnock	•••	30	Thatcher	•••

Abbildung 5.6: Keywords, die im BNC häufiger vorkommen als im enTenTen13 Korpus

in Korpus B und größer eins bis unendlich für eine häufigere Verwendung in Korpus A. Bei N handelt es sich um eine Korrekturkonstante, die dafür sorgt, dass entweder seltene oder hochfrequente Wörter höher gewichtet werden. Die Darstellung der Ergebnisse erfolgt als sortierte Liste, wobei die Sortierung nicht umgekehrt werden kann. Um die Richtung des Vergleichs zu ändern, müssten Korpus und Referenzkorpus getauscht werden. Durch die fehlenden Werte der Keyness kann jedoch nicht abgeschätzt werden, wie stark die Überbenutzung im Vergleich zum Referenzkorpus ausfällt. Außerdem findet kein Test auf statistische Signifikanz statt. Im Gegensatz zu anderen Terminologieextraktionsverfahren (Chowdhury, Gliozzo & Trewin 2018, Alrehamy & Walker 2017, Amjadian, Inkpen, Paribakht & Faez 2016) erscheint dieses Vorgehen oberflächlich.

Mit SketchEngine kann ebenso ein „One-Click Dictionary" erstellt werden. Hierbei werden die Ergebnisse aus SketchEngine in ein XML Format übertragen und an Lexonomy[24] übergeben (Měchura 2017). Der Mehrwert von Lexonomy ergibt sich aus der automatisierten Formatierung der Ergebnisse. Anstatt selbst aus den Ergebnissen einer Suche, aus N-Grams oder aus Keywords Wörterbucheinträge zu formulieren und mit Beispielsätzen zu versehen, wird in Lexonomy nur das Schema der Wörterbucheinträge definiert und die Schnittstelle konfiguriert. Danach kann mit Hilfe der oben beschriebenen Funktionen nahezu der komplette Eintrag automatisiert erzeugt werden (Měchura 2017).

SketchEngine ist eines der wenigen online Konkordanzwerkzeuge[25], die es erlauben, eigene Texte zu analysieren. Da SketchEngine auf Servern gehostet wird, ist die Suche unabhängig von der Hardware der suchenden Person. Daher können auch große Korpora innerhalb weniger Sekunden durchsucht werden. Ebenso ist der Funktionsumfang deutlich größer als bei den bisher diskutierten Konkordanzanwendungen. Allerdings müssen einige Punkte beim Einsatz in

[24] Bei Lexonomy handelt es sich um ein Onlineangebot zur automatisierten Erstellung von Wörterbüchern.
[25] Im Gegensatz etwa zum BNC oder dem COCA auf der Seite der Brigham Young University, USA.

Schreibberatungen bedacht werden.

Auch wenn bei der Dokumentation der Funktionen von SketchEngine vielfach eine einfache Erklärung und linguistische Fachartikel vorhanden sind, werden sowohl die Funktionen als auch die Einstellungsmöglichkeiten mit linguistischen Fachbegriffen benannt. Eine deutsche Übersetzung der Software ist nur lückenhaft vorhanden, sodass nur ein Teil des Frontends übersetzt wird. Für Schreibberater_innen und Studierende mit geringen Englischkenntnissen könnte dies die Nutzbarkeit einschränken.

Je nach pädagogischem Konzept der Schreibberatung fallen unterschiedlich hohe Kosten für die Nutzung von SketchEngine an. Die Kosten ergeben sich aus der Anzahl an Nutzenden und der Größe der eigenen auf SketchEngine hochgeladenen Korpora. Wenn Studierende SketchEngine außerhalb der Schreibberatung nutzen wollen, müssen zusätzliche Accounts gekauft werden. Zusätzlich wird für Studierende je nach Studienort ein anderer Preis verlangt.

Der gravierendste Nachteil von SketchEngine in Hinblick auf das hier beschriebene Einsatzszenario ist jedoch, dass keinerlei Aufbereitungsmöglichkeiten für Texte angeboten werden. Dies bedeutet, dass ein zweites Werkzeug notwendig ist, um, wie in der Ausgangssituation dieser Arbeit beschrieben, wissenschaftliche Texte spezifischer Fachrichtungen zu einem Korpus zusammenzustellen. Die Komplexität und die technischen Anforderungen erhöhen sich dementsprechend.

Außerdem bleibt das juristische Problem bestehen, dass nicht klar ist, ob das Konvertieren und Hochladen auf einen fremden Server von urheberrechtlich geschützten Werken zu Bildungszwecken erlaubt ist.

5.2.8 Zusammenfassung

Die letzten Kapitel haben eine Auswahl an möglichen linguistischen Programmen zur Unterstützung in Schreibberatungen aufgezeigt und beschrieben. Dabei wurde davon ausgegangen, dass eigene Texte analysiert werden sollen, diese Texte einer Aufbereitung bedürfen und sowohl Student_innen als auch Schreibberater_innen wenig Berührungspunke mit Korpuslinguistik in Verbindung mit den dazugehörigen Programmen vorweisen können. Diese Annahmen sind für den Erkenntnisgewinn dieser Arbeit essentiell.

Tabelle 5.1 fasst die Charakteristika der bisher diskutierten Programme zusammen. Zusätzlich werden der Vollständigkeit halber in Tabelle 5.2 anhand derselben Kategorien bisher nicht diskutierte reine online Werkzeuge mit fokussiertem Funktionsumfang (Coh-Metrix/BNC & COCA) und linguistische Pakete für populäre Programmiersprachen (R/Python) vorgestellt.

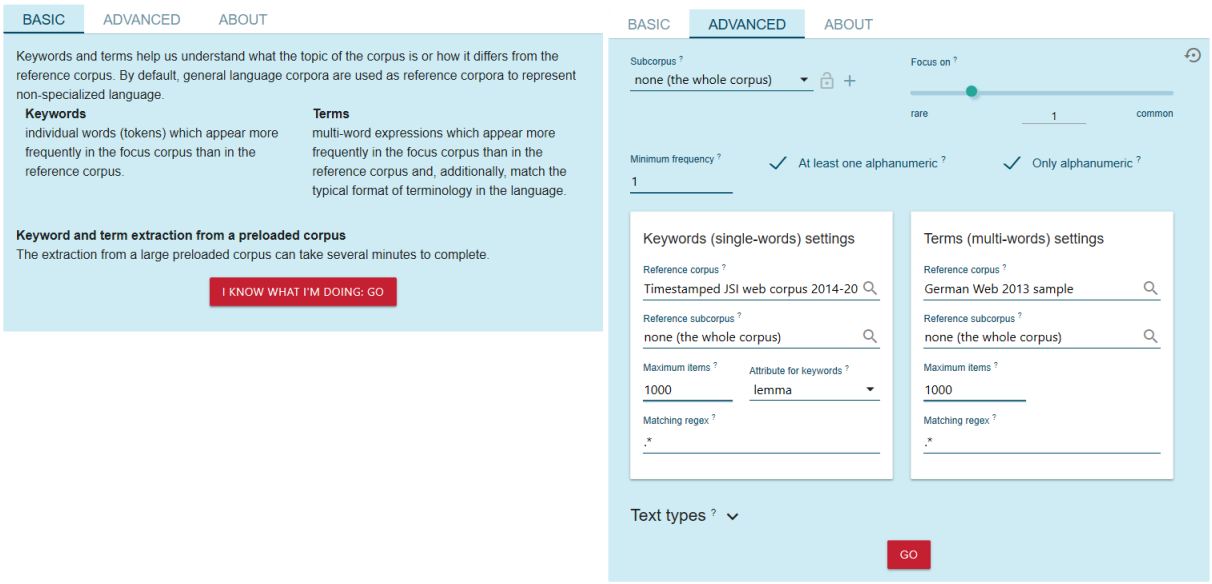
Die betrachteten Charakteristika sind in vier Kategorien eingeteilt: Die linguistischen Funktionen umfassen sowohl die Suchmöglichkeiten als auch die Ergebnistypen der Programme. Da diese Programme in einer Schreibberatung eingesetzt werden sollen, beschreiben die didaktischen Funktionen die Anpassbarkeit an die Bedürfnisse von linguistischen Laien. Die technischen Kategorien zeigen die Verwendungsmöglichkeit auf unterschiedlichen Betriebssystemen, ob der Quellcode offen liegt und daher angepasst werden kann und ob die Programme auch mit größeren Datenmengen funktionieren. Abschließend werden die minimalen Kosten für eine Li-

zenz aufgezeigt. Bei allen Programmen bezieht sich die Auswertung auf den aktuellsten Stand vom 31. März 2020.

Anhand der Charakteristika können einige Programme zu Gruppen zusammengefasst werden. Die Erweiterungen von R, Python und Java weisen eine hohe Komplexität und einen hohen Innovationsgrad in Bezug auf ihre Funktionen und Algorithmen. Allerdings verfügen sie nicht über eine grafische Oberfläche, die es auch Studierenden und Schreibberater_innen ohne IT Kenntnisse und korpuslinguistische Fähigkeiten erlauben würde, sie zu nutzen. CorpKit, WordStatix und ShinyConc basieren unter anderem auf den Technologien der vorherigen Gruppe, jedoch sind sie in ihrem Funktionsumfang auf einzelne Anforderungen beschränkt. Sie bilden das Bindeglied zu einer weiteren Gruppe, welche aus AntConc und WST besteht. Diese beiden Programme verfügen zwar über eine grafische Oberfläche, sind aber technisch insofern beschränkt, als dass sie nicht über die Grenzen des eingesetzten Computers hinaus skalieren. Außerdem können die Funktionen nicht angepasst und erweitert werden. SketchEngine, Coh-Metrix und das BNC/COCA Interface erlauben ebenso wenig eine Anpassung der Funktionen und Algorithmen. Jedoch werden sie als online Programme angeboten, sodass keine eigene Laufzeitumgebung notwendig ist. Bis auf WST und SketchEngine sind alle Werkzeuge kostenlos. Während das Lizenzmodell von WST aus unpersonalisierten Lizenzen besteht, die lokal bei der Nutzung von WST hinterlegt werden, basiert das Lizenzmodell von SketchEngine auf personalisierten Nutzer_innenlizenzen. Hierbei sind die Kosten vom Status der Anwendender_innen abhängig, sodass der Preis zwischen Studierenden und kommerziell Nutzenden variiert, sowie von dem Land der Nutzenden, der Größe der eingespeisten Korpora und den Zahlintervallen abhängt.

Alle Werkzeuge richten sich primär an Linguist_innen. Dementsprechend wird Fachvokabular nicht in den Anwendungen erklärt, sondern gegebenenfalls auf die jeweilige Dokumentation und Veröffentlichung verwiesen. Einzig SketchEngine ermöglicht es, die Komplexität der Funktionen, die über KWIC hinausgehen, in mehreren Stufen an das eigene Können und Wissen anzupassen. Abbildung 5.7 verdeutlicht am Beispiel der „Keyword" Funktion von SketchEngine dieses Vorgehen. Es stehen zwei Komplexitätsstufen zur Verfügung: In der „Basic" Konfiguration wird zwar die Idee von Keywords und Terms erläutert, allerdings bestehen hier keine Einstellungsmöglichkeiten. Die „Advanced" Konfiguration hingegen ermöglicht eine Feinjustierung der Funktion. Während die „Basic" Darstellung noch ohne erklärende Pop-Ups ausgekommen ist, befindet sich bei der „Advanced" Konfiguration hinter jeder Einstellung eine Erklärung. Diese Erklärungen wiederum enthalten Verknüpfungen zur erweiterten Dokumentation der Fachbegriffe. Die 15 mit „?" markierten Erklärungen in Abbildung 5.7b verweisen auf insgesamt 26 weitere Erklärungsseiten. Somit wird nicht die Komplexität reduziert, um das Werkzeug einsteigerfreundlicher zu gestalten, sondern eine möglichst umfassende Dokumentation für fortgeschrittene Nutzer_innen angeboten.

Insgesamt zeigt sich, dass keines der hier diskutierten Programme darauf ausgerichtet ist, fortschrittliche Analysemethoden für fachfremde Nutzer_innen zur Verfügung zu stellen. Denn sowohl die Suchmasken und Einstellungsmöglichkeiten als auch die Ergebnisse werden nicht an den Bedürfnissen dieser Nutzer_innengruppe ausgerichtet. Dadurch bedürfen solche Werkzeu-



(a) „Basic Keywords" Funktion (b) „Advanced Keywords" Funktion

Abbildung 5.7: Screenshots der „Keywords" Funktion in SketchEngine

ge, wenn sie in einer Schreibberatung eingesetzt werden sollen, einerseits (computer-)linguistisch geschultes Personal und andererseits ein didaktisches Vorgehen, dass entweder die Studierenden umfassend in die Funktionsweise der Programme einweist oder sie durch ihre fehlende Expertise an die Unterstützung durch die Schreibberater_innen bindet.

5.2.9 Addendum: Alternativen zu R auf Basis des Apache Stacks

Bisherige korpuslinguistische Programme wie auch HanConc nutzen vielfach R und R Shiny als technologische Basis. Jedoch gibt es bestehende Lösungen, die R in Bezug auf Funktionalität und Performance bei weitem übertreffen. Zu Recht kann eingewendet werden, dass R als Skriptsprache für statistische Verfahren entwickelt wurde (R Core Team 2021) und weniger als vollwertige allgemeine Programmiersprache. Daher gibt es auch etwa Programme und Bibliotheken auf Basis von Java, die speziell für den großflächigen Einsatz als Webserver oder Volltextsuchmaschine mit großen Datenmengen entwickelt wurden. Anhand verschiedener Komponenten der Apache Software Foundation wird beispielhaft eine Alternative aufgezeigt, die schneller, stabiler und skalierbarer als HanConc zum jetzigen Zeitpunkt ist.

Mit etwa Solr, Hadoop, Tomcat, Cassandra, Play und Swing stehen alle notwendigen Komponenten zur Verfügung, um HanConc mit Open Source Software aus der Apache Familie zu ersetzen (Estrada & Ruiz 2016). Eine Alternativarchitektur würde so aussehen, dass Solr die Dissertationen indexieren und als Suchmaschine fungieren würde. Die Daten würden auf einem Hadoop Cluster gespeichert werden. Die Nutzer_inneneingaben und die Programmausgaben könnten über Play auf einem Tomcat Webserver oder über Swing als eigenständige Desktop-App erfolgen. Wie in Abbildung 5.8 gezeigt, könnte die gesamte Anwendung in Java und/oder Scala geschrieben werden. Im Vergleich zu HanConc kommt dieses Szenario mit weniger Komponenten aus (siehe Kapitel 5.3). Dies liegt vor allem daran, dass mit Solr die Textaufbereitung

Tabelle 5.1: Charakteristika der bisher diskutierten linguistischen Anwendungen (I)

Programme	WordSmith Tools	AntConc	CorpKit	WordStatix	ShinyConc
Linguistische Funktionen:					
KWIC	✓	✓	✓	✓	✓
Kollokationen	✓	✓	✓	x	x
N-Grams	✓	✓	✓	x	x
Wortlisten	✓	✓	x	x	x
Frequenzen pro Text	✓	✓	x	✓	✓
Statistische Auswertungen	✓	✓	x	x	x
Thesaurus/Wortassoziationen	x	x	x	x	x
Konkordanzplot	✓	✓	x	✓	x
Grafiken	✓	✓	✓	✓	x
Vergleich mehrerer Korpora	x	x	x	x	✓
Mehrwortsuche	✓	✓	✓	x	✓
Kombinierte Suche	x	x	✓	x	x
RegEx	x	✓	✓	x	✓
Eigene Texte	✓	✓	✓	✓	✓
Textaufbereitungsfunktionen	x	x[a]	x	x	x
Unterstützung für deutsche Texte	✓	✓	✓	✓	✓
Didaktische Funktionen:					
Vereinfachte linguistische Sprache	x	x	x	x	x
Skalierbarkeit der Komplexität	x	x	x	x	x
Technische Eigenschaften:					
Grafisches Frontend	✓	✓	✓	✓	✓
Windows	✓	✓	✓	✓	✓
Linux	x	✓	✓	✓	✓
Mac	x	✓	✓	✓	✓
Server	x	x	✓	x	✓
Open Source	x	x	✓	✓	✓
Benötigt Programmierkenntnisse	x	x	✓	x	✓
Lauffähig für größere Korpora	✓	✓	x	x	✓
Minimale Kosten:	£50	Kostenlos	Kostenlos	Kostenlos	Kostenlos

[a] Über andere Tools

Tabelle 5.2: Charakteristika der bisher diskutierten linguistischen Anwendungen (II)

Programme	SketchEngine	Coh-Metrix	BNC/COCA	tm/corpustools	NLTK
Linguistische Funktionen:					
KWIC	✓	x	✓	x	✓
Kollokationen	✓	x	x	x	✓
N-Grams	✓	x	x	x	✓
Wortlisten	✓	x	x	✓	✓
Frequenzen pro Text	x	x	✓	✓	✓
Statistische Auswertungen	x	✓	x	✓	✓
Thesaurus/Wortassoziationen	✓	x	x	✓	✓
Konkordanzplot	x	x	x	x	x
Grafiken	x	x	x	✓	✓
Vergleich mehrerer Korpora	✓	x	x	✓	✓
Mehrwortsuche	✓	x	✓	x	✓
Kombinierte Suche	✓	x	✓	x	✓
RegEx	✓	x	✓	x	✓
Eigene Texte	✓	x	x	✓	✓
Textaufbereitungsfunktionen	x	x	x	✓	✓
Unterstützung für deutsche Texte	✓	x	x	✓	✓
Didaktische Funktionen:					
Vereinfachte linguistische Sprache	x	x	x	x	x
Skalierbarkeit der Komplexität	✓	x	x	x	x
Technische Eigenschaften:					
Grafisches Frontend	✓	✓	✓	x	x
Windows	x	x	x	✓	✓
Linux	x	x	x	✓	✓
Mac	x	x	x	✓	✓
Server	x	x	x	x	✓
Open Source	x	x	x	✓	✓
Benötigt Programmierkenntnisse	x	x	x	✓	✓
Lauffähig für größere Korpora	✓	✓	✓	✓	✓
Minimale Kosten:	69€ p.a.	k.a.	kostenlos	kostenlos	kostenlos

und -suche in einer Komponente vereint ist. Mit Play und Swing steht auch hier die Möglichkeit offen, die entwickelte Software Plattform unabhängig als Desktop-Applikation oder online einzusetzen.

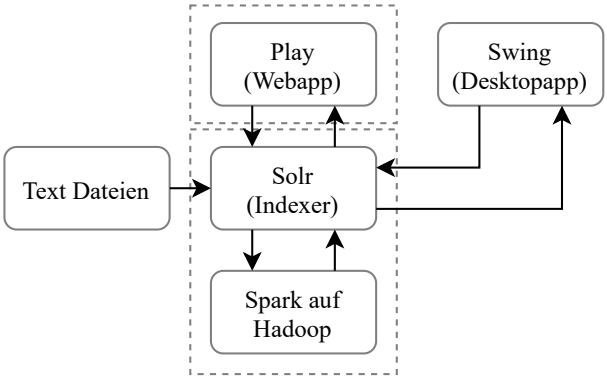


Abbildung 5.8: Vereinfachte alternative Architektur mit Solr

Im Vergleich zu Abbildung 5.8 wurde Solr in Abbildung 5.9 durch eine Datenbankkomponente, in diesem Fall Cassandra, ersetzt. Hierdurch könnten mehr Funktionen von HanConc genauer umgesetzt werden. So könnte etwa der PoS Tagger freier gewählt werden, da er nicht mit Solr kompatibel sein muss. Die Textsuche wäre wieder eine einfache Datenbankabfrage.

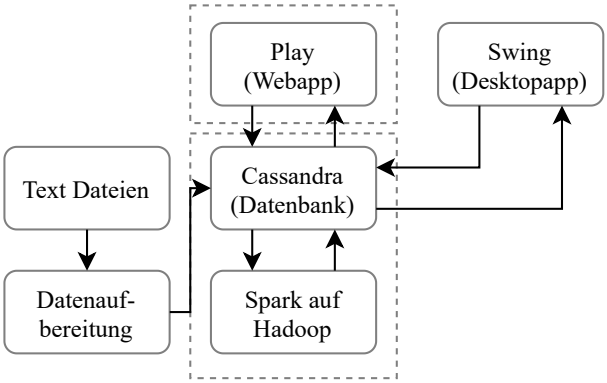


Abbildung 5.9: Vereinfachte alternative Architektur mit Datenbank

Es ist an dieser Stelle zu bedenken, dass die hier beschriebenen Komponenten auf mindestens einem wahrscheinlich jedoch eher auf mehreren Servern installiert werden müssten.

Matthijs Brouwer und Marc Kemps-Snijders beschreiben in ihrem Artikel eine Erweiterung zu Solr und Lucene (2012), die den Einsatz von PoS Taggern als Annotationsmethode und Sucherweiterung ermöglicht. Sie listen die bereits in Kapitel 5.2 diskutierten Lösungen auf, wodurch ein Dilemma mit Solr deutlich wird. Denn trotz der offensichtlichen technischen Vorteile ist die oben beschriebene Lösung für den Einsatz in Schreibzentren und als gemeinsame Entwicklungsumgebung für Linguist_innen ungeeignet.

Auch wenn das gesamte System quelloffen ist, ist es dennoch nicht für Einsteiger_innen geeignet. Für die in den Abbildungen 5.8 und 5.9 dargestellten Komponenten sind etwa folgende Softwarepakete und Sprachen zu erlernen:

Tabelle 5.3: Minimal notwendige zu erlernende Komponenten und Programmiersprachen für die skizzierten Lösungen im Vergleich zu HanConc

Alternative Lösungen basierend auf Apache Lösungen		HanConc	
Softwarepakete	Sprachen	Softwarepakete	Sprachen
Play[26]	Java	RStudio	R
Swing[27]	Scala		
Spark	Cassandra Query Language[28]		
Hadoop[29]			
Maven			
Scala Building Tool[30]			
Solr[31]			
Cassandra[32]			

Wird ein Webfrontend verwendet, so sind beide Seiten der Tabelle noch um HTML und CSS zu erweitern. Dennoch sind für eine R basierte Lösung wie HanConc deutlich weniger Komponenten notwendig, um eine minimal lauffähige Lösung für die in Kapitel 5.1 aufgestellten Anforderungen zu entwickeln. Vor dem Hintergrund, dass HanConc von Schreibberater_innen und Linguist_innen ohne Hilfe von Programmierer_innen und Informatiker_innen weiterentwickelt und erweitert werden soll, ist eine geringe Einstiegshürde jedoch essentiell.

Lösungen, die eine zu hohe Einstiegshürde haben, führen dazu, dass pädagogische Konzepte eventuell nicht umgesetzt werden können. So wäre es beispielsweise denkbar, zusammen mit Student_innen HanConc weiterzuentwickeln und gemeinsam anzupassen. Ist eine häufige Hilfestellung notwendig, weil die Linguist_in oder Schreibberater_in nicht selbstständig programmieren kann, wird die Zusammenarbeit mit den Studierenden unnötig verkompliziert oder gar komplett verhindert.

Anwendungen mit Solr als Suchengine beruhen auf einfachen Suchen mit einem oder mehreren Strings, die möglichst schnell aus tendenziell riesigen Textmengen einen Dokumentausschnitt mit dem Suchbegriff inklusive seiner Umgebung und den Link zum Dokument zurückgeben. Die Idee hinter HanConc ist jedoch, eine deutlich komplexere Suche innerhalb von Texten, die auf mehreren Ebenen annotiert sind, und eine detaillierte Analyse der Ergebnisse zu ermöglichen.

Aufgrund der noch viel größeren technischen Komplexität wird auf eine Diskussion von Java, JavaScript oder PHP basierten 3-Tier Webanwendungen verzichtet.

Es ist festzuhalten, dass es kostenlose Open Source Lösungen gibt, die in Kombination durchaus eine Alternative zu HanConc sein könnten. HanConc soll jedoch vor allem die Ar-

[26] Nicht notwendig, wenn nur ein Desktopfrontend vorgesehen ist
[27] Nicht notwendig, wenn nur ein Webfrontend vorgesehen ist
[28] Nicht notwendig, wenn Solr anstatt Cassandra vorgesehen ist
[29] Nicht zwangsläufig notwendig, wenn keine verteilte Datenhaltung vorgesehen ist
[30] Nicht notwendig, wenn die Lösung nur in Java implementiert wird
[31] Nicht notwendig, wenn eine Datenbankarchitektur vorgesehen ist
[32] Nicht notwendig, wenn Solr anstatt Cassandra vorgesehen ist

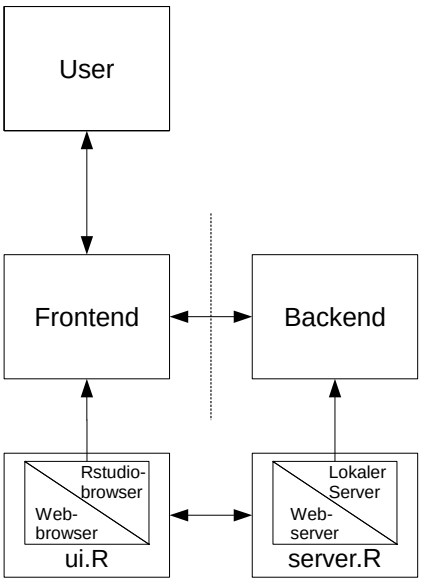


Abbildung 5.10: Architektur von HanConc

beit mit Korpora in Schreibberatungen vereinfachen und verbessern. Daher ist eine besonders elegante oder ausgefeilte Programmierung und IT Architektur hierfür nicht notwendig, solange HanConc noch eine akzeptable Performanz bietet. Es wird deshalb eher auf eine niedrige Einstiegshürde bei der Installation und Wartung, eine möglichst hohe Portabilität zwischen Betriebssystemen, niedrige Hardwareanforderungen und eine möglichst einfache Veränderbarkeit durch Interessierte mit wenig Programmiererfahrung Wert gelegt.

5.3 Teilbereiche von HanConc

Bei HanConc handelt es sich um eine Webapplikation, bei der die Benutzereingaben (siehe Kapitel 5.3.7) in die Maske der Webseite (Frontend) eingetragen und an den Server (Backend) übergeben werden, der dann Berechnungen anstellt und das Ergebnis an das Frontend zurückgibt (siehe Kapitel 5.3.8). Sowohl das Frontend als auch das Backend können lokal auf einem Computer oder auf einem Webserver ausgeführt werden.

Abbildung 5.10 verdeutlicht den Funktionsaufbau von HanConc. Der User kommuniziert in HanConc mit dem R Server über das Frontend, das in der $ui.R$ Datei beschrieben wird. Dieser R Server kann lokal auf einem Computer oder auf einem Server gehostet werden und führt auf Basis der Benutzereingaben auch alle Berechnungen durch. Alle Funktionen werden in der $server.R$ Datei beschrieben[33]. Beide Dateien kommunizieren ihre Anfragen und Ergebnisse über eine Liste. Dabei können sowohl die Anfragen als auch die Berechnungen und Ergebnisse beliebig komplex gestaltet werden.

Die nächsten Kapitel sind wie folgt strukturiert: Zunächst sollen einige Screenshots einen Eindruck von HanConc liefern. Die allgemeine Software-Architektur von HanConc wird in

[33]Um eine hohe Qualität und Wartbarkeit des Quellcodes zu gewährleisten, wird der Code auf mehrere Funktionen aufgeteilt, welche wiederum auf mehrere Dateien verteilt werden.

Kapitel 5.3.2 aufgezeigt. Kapitel 5.3.3 erläutert, wie die Rohtexte aufbereitet und in HanConc eingespeist werden. Kapitel 5.3.4 zeigt an einem Beispiel, wie HanConc um Zusatzfunktionen ergänzt werden kann. Als grundsätzlich andere Herangehensweise an Texte wird in Kapitel 5.3.5 gezeigt, wie aus syntaktisch strukturierten Texten Matrizen erstellt werden. Kapitel 5.3.6 erläutert im Detail, wie HanConc auf verschiedenen Systemen gehostet werden kann. In Kapitel 5.3.7 wird beschrieben, wie der Userinput über das Frontend aufgenommen und an das Backend weitergeleitet wird. Als zentrales Kapitel diskutiert Kapitel 5.3.8 die Textanalysefunktionen von HanConc und wie diese in Schreibberatungen eingebettet werden können. Kapitel 5.3.9 zeigt abschließend Wege auf, wie HanConc kontinuierlich verbessert werden kann. In den Unterkapiteln wird gegebenenfalls darauf eingegangen, wie HanConc an die (möglichen) User angepasst werden kann (siehe Kapitel 3).

5.3.1 Frontend von HanConc

Das Frontend von HanConc besteht aus R Shiny Widgets für grundlegende Funktionalitäten wie Buttons, Slider oder Textinputfelder. Ebenso werden mit diesen Widgets rudimentäre HTML Anpassungen wie ein Banner der LUH, Kontaktinformationen und einige CSS Veränderungen gegenüber den Shiny Standards ermöglicht, um etwa die Corporate Identity Farben der LUH aufzugreifen.

Abbildung 5.11 zeigt HanConc zum Startzeitpunkt[34]. Um Nutzer_innen einen niedrigschwelligen Einstieg zu ermöglichen, werden einige Funktionen ausgeblendet. Erst durch auswählen der „Advanced Query" Checkbox werden weitere Elemente angezeigt. Die eigentliche Suche wird mit „Update" gestartet und die Ergebnisse werden im unteren Teil des Bildschirms angezeigt. Je nach Sucheinstellungen werden tabellarische Ergebnisse unter dem „KWIC"- oder den jeweiligen „Graph"-Reitern angezeigt.

Um einen genaueren Eindruck von HanConc zu vermitteln, wird auf Abbildung 5.12 verwiesen. Abbildung 5.12(a) zeigt eine einfache Suche und deren Ergebnisse. Hier wird in einem Korpus nach einem Wort und nur nach KWIC gesucht[35]. Um dieses Ergebnis zu generieren sind nur vier Klicks und eine Eingabe notwendig. Damit ist die Suche kaum komplexer als bei modernen Internetsuchmaschinen.

Abbildung 5.12(b) zeigt die Auswahloptionen für eine komplexe Suche. Entsprechend der Erkenntnis, dass die Fakultätszugehörigkeit ein valides Unterscheidungskriterium ist (siehe Kapitel 4), wurde hier die Möglichkeit geschaffen, das Korpus mit wenigen Klicks zu wechseln oder zwei Korpora miteinander zu vergleichen. Durch die Auswahl zweier Korpora werden die Suchparameter auf beide Korpora angewendet.

Anstatt eines einzelnen Wortes wird bei diesem Beispiel für eine komplexe Suche unter

[34] Die Überschrift der Webseite ergibt sich aus den Wünschen der damaligen Stakeholder, die ein auf Ingenieurwissenschaften reduziertes Korpus für die initiale Bereitstellung und Präsentation des Programms angefordert hatten. Ebenso wurde das Frontend mit englischen Beschriftungen programmiert. Zu einer späteren deutschsprachigen Umsetzung ist es nicht gekommen.

[35] Die weiteren Ergebnistypen, die über einfache KWIC hinausgehen und sich hinter der Auswahlfläche „Result Types" verbergen, werden in späteren Unterkapiteln besprochen. Dort wird auch auf die jeweilige Darstellung im Frontend eingegangen.

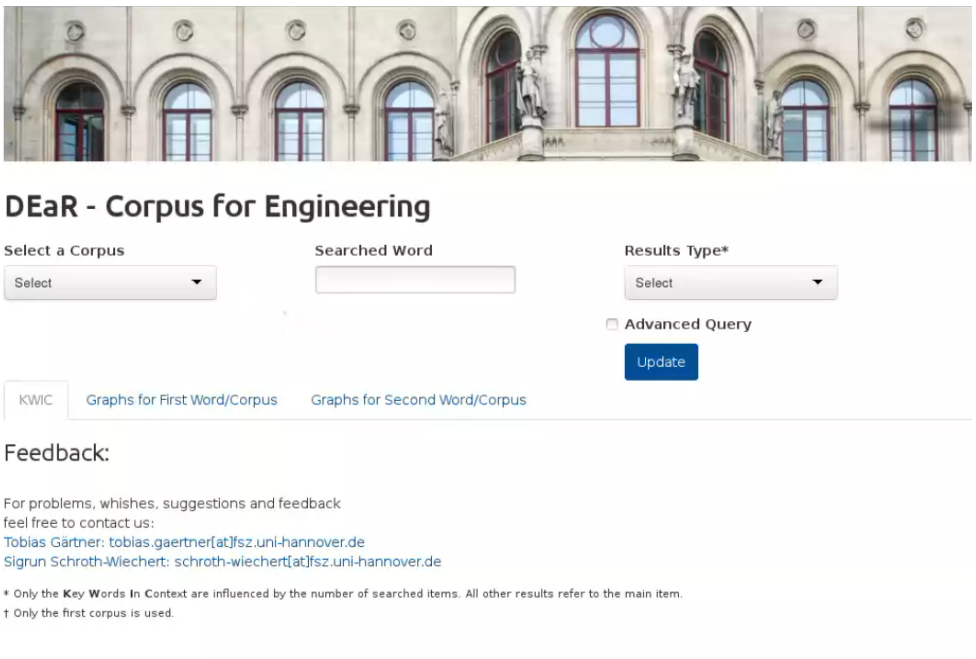


Abbildung 5.11: Startansicht von HanConc

Berücksichtigung der Reihenfolge der Suchparameter[36] nach der Kombination aus „Methode", „zur" und einem normalen Nomen[37] gesucht. Die Ergebnisse ähneln denen aus Abbildung 5.12(a) mit dem Unterschied, dass entsprechend mehr Worte markiert werden.

Um den Umfang dieser Arbeit nicht unnötig auszuweiten, wird auf eine detaillierte Beschreibung der optischen Merkmale von HanConc verzichtet. Stattdessen sei auf das Repository, in dem der Quellcode von HanConc liegt, verwiesen. Dort befindet sich ein Video, das alle Funktionen von HanConc an Beispielen zeigt. Außerdem wird bei der Diskussion der Ergebnistypen (siehe Kapitel 5.3.8) auch ausgeführt, wie diese im Frontend dargestellt werden.

5.3.2 Softwarearchitektur von HanConc

Die Grundfunktionen von HanConc sind in R geschrieben und im Shiny Framework implementiert[38]. Entsprechend dieses Frameworks besteht HanConc aus einer $server.R$ und einer $ui.R$ Datei. Diese beiden Dateien bilden das Grundgerüst für das Front- und Backend. Alle weiteren Funktionen, die in späteren Kapiteln beschrieben werden, werden über diese Dateien aufgerufen. Um HanConc zu starten, muss das Arbeitsverzeichnis der aktuellen R Sitzung auf den Ordner mit den beiden R Dateien zeigen und dort die runApp Funktion des Shiny Pakets ausgeführt werden[39]. Diese Funktion startet den Webserver, baut aus den Beschreibungen der

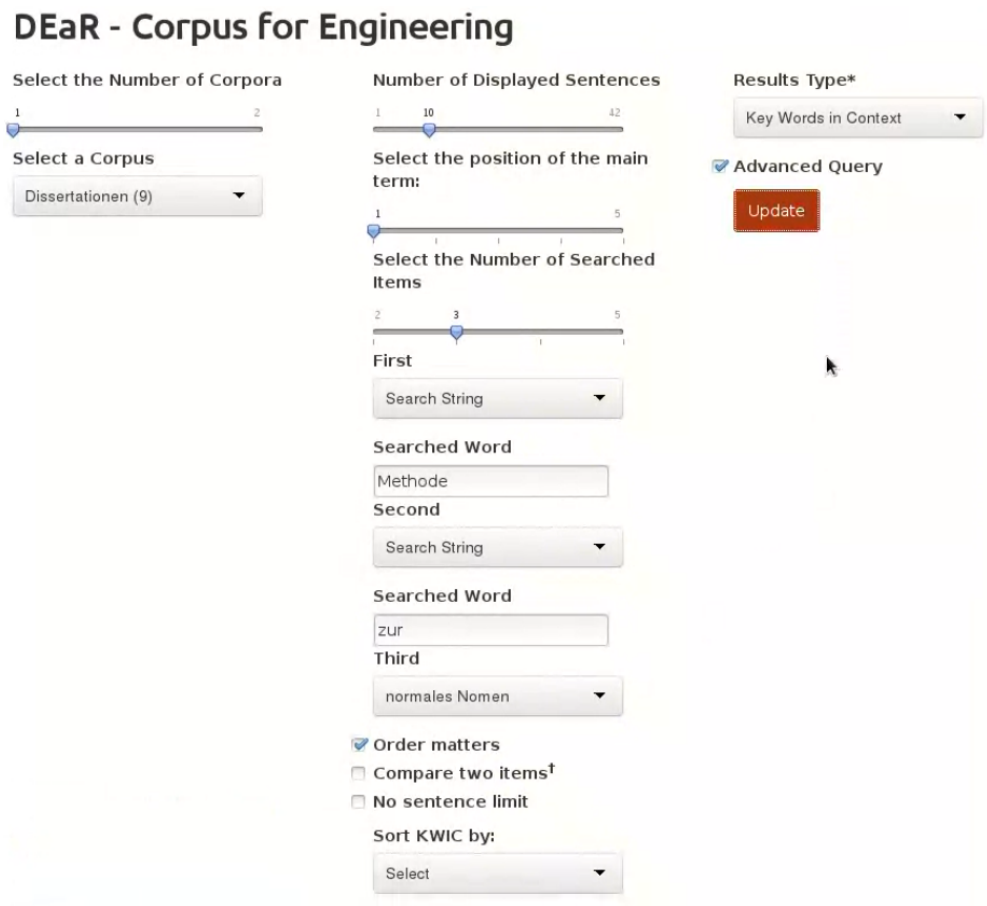
[36] Durch die Suchoption „Order matters" werden nur Sätze zurückgegeben, die „Methode" + „zur" + NN als PoS Tag enthalten. Wird diese Suchoption nicht ausgewählt, gibt HanConc auch Sätze zurück, in denen weitere Elemente zwischen den Suchelementen stehen und/oder solche Sätze, in denen diese Suchelemente in einer anderen Reihenfolge verwendet werden.

[37] In HanConc und dieser Arbeit wird das Stuttgart-Tübingen-Tagset (STTS) verwendet.

[38] https://shiny.rstudio.com/reference/shiny/1.1.0/ Die Versionsnummer ist durch die aktuelle zu ersetzen. (Stand: 10. März 2020)

[39] Für Endanwender_innen befindet sich im Hauptverzeichnis von HanConc für Windows eine $startHanConc.bat$ und für Linux eine $startHanConc.sh$ Datei, die es je nach Betriebssystem ermöglicht,

(a) Tabellarische Beispielergebnisse einer einfachen Suche



(b) Auswahloptionen einer komplexen Suche

Abbildung 5.12: Beispiele für einfache und komplexe Suchen mit HanConc

$ui.R$ Datei das Web Frontend und aus der $server.R$ Datei das Backend der Applikation. Die eigentliche Serverengine beruht auf Node.js in der Version 12[40].

HanConc mit nur einem Klick zu starten.

[40] https://rstudio.com/products/shiny/download-server/ (Stand: 10. März 2020)

Abbildung 5.13 zeigt die Funktionsaufrufe in HanConc[41]. Dargestellt sind nur die originären Funktionen von HanConc. Funktionen aus anderen Paketen sind an dieser Stelle aus Gründen der Übersichtlichkeit nicht dargestellt.

HanConc erweitert Standardfunktionen von Shiny in zwei Richtungen. Vor die runApp Funktion wird eine Wrapper Funktion gesetzt. Diese $initialise$ genannte Funktion lädt die zusätzlichen R Dateien von HanConc in die aktuelle R Sitzung, stellt sicher, dass die notwendigen R Pakete geladen sind, ruft die runApp Funktion auf und übergibt ihr den Speicherort von HanConc. Mit Hilfe dieser Konstruktion muss nur eine einzige R Datei und die darin enthaltene Funktion mit einem einzigen Parameter, dem Speicherort von HanConc, aufgerufen werden. Da dieser Aufruf direkt aus dem HanConc Ordner erfolgt, muss dieser Parameter von der Benutzer_in nicht übergeben werden. Um eine Lösung zu erreichen, bei der die Benutzer_in nur auf eine einzige Datei klicken muss, erfolgt der Aufruf der $initialise$ Funktion je nach Betriebssystem über die $startHanConc.bat$ Datei für Windows oder die $startHanConc.sh$ Datei für Linux. Hierbei wird R über die jeweilige Kommandozeile gestartet, die $initialise.R$ Datei eingelesen und die $initialise$ Funktion aufgerufen. Als Parameter wird der Speicherort der bat bzw. sh Datei übergeben. Dieses Vorgehen soll vor allem das erstmalige Starten von HanConc erleichtern, da so verhindert wird, dass HanConc nur auf Grund einer Fehlkonfiguration oder durch ungünstig gesetzte Startparameter nicht startet.

Die für eine Schreibberatung relevanten Funktionen werden aus der $server.R$ Datei aufgerufen. Die enthaltene Server Funktion ruft drei Kategorien von Funktionen auf: Suchfunktionen, Grafikfunktionen und solche, die die Suchergebnisse linguistisch und statistisch aufbereiten. Die Suchfunktionen ($corpus.to.sentences$ und $more.than.one.item$) reduzieren das Korpus auf solche Sätze, die den Suchparametern entsprechen. Die der $corpus.to.sentences$ Funktion nachgelagerten Funktionen erweitern das Ergebnis um die umliegenden Sätze ($largerContext$) und zusätzliche Textaufbereitungen für die Lesartenfunktion ($createWordVectorMatrix$; siehe Kapitel 5.3.8.9). Wird mit mehr Parametern als nur einem Wort gesucht, so wird die $more.-than.one.item$ Funktion auf die Ergebnisse der $corpus.to.sentences$ Funktion angewandt, um diese entsprechend der zusätzlichen Suchparameter einzuschränken. Sowohl die $results$ Funktion und ihre nachgelagerten Funktionen als auch die $model.topic$ und die $toWordSenses$ Funktion berechnen je nach Nutzereingabe auf Basis der Suchergebnisse linguistische und statistische Kennzahlen. Bei $hit.per.text.hist$, $position.hit.per.text.hist$ und $wordCloud$ handelt es sich um Funktionen, die eine grafische Aufbereitung der Ergebnisse der oben genannten Funktionen vornehmen. Die $set.corpus$ Funktion stellt aus den geladenen Korpora, Metadaten der Korpora und Term-Dokument Matrizen der Korpora den Suchraum zusammen und übergibt ihn an die Suchfunktionen.

5.3.3 Textaufbereitung für HanConc

Die Textgrundlage für die hier vorgestellten Korpora sind die digital zur Verfügung stehenden Dissertationen der TIB/UB der LUH. Die Promotionsordnungen schreiben zwingend die Ab-

[41] Zur besseren Lesbarkeit wurden die Funktionsnamen in der Abbildung umformatiert.

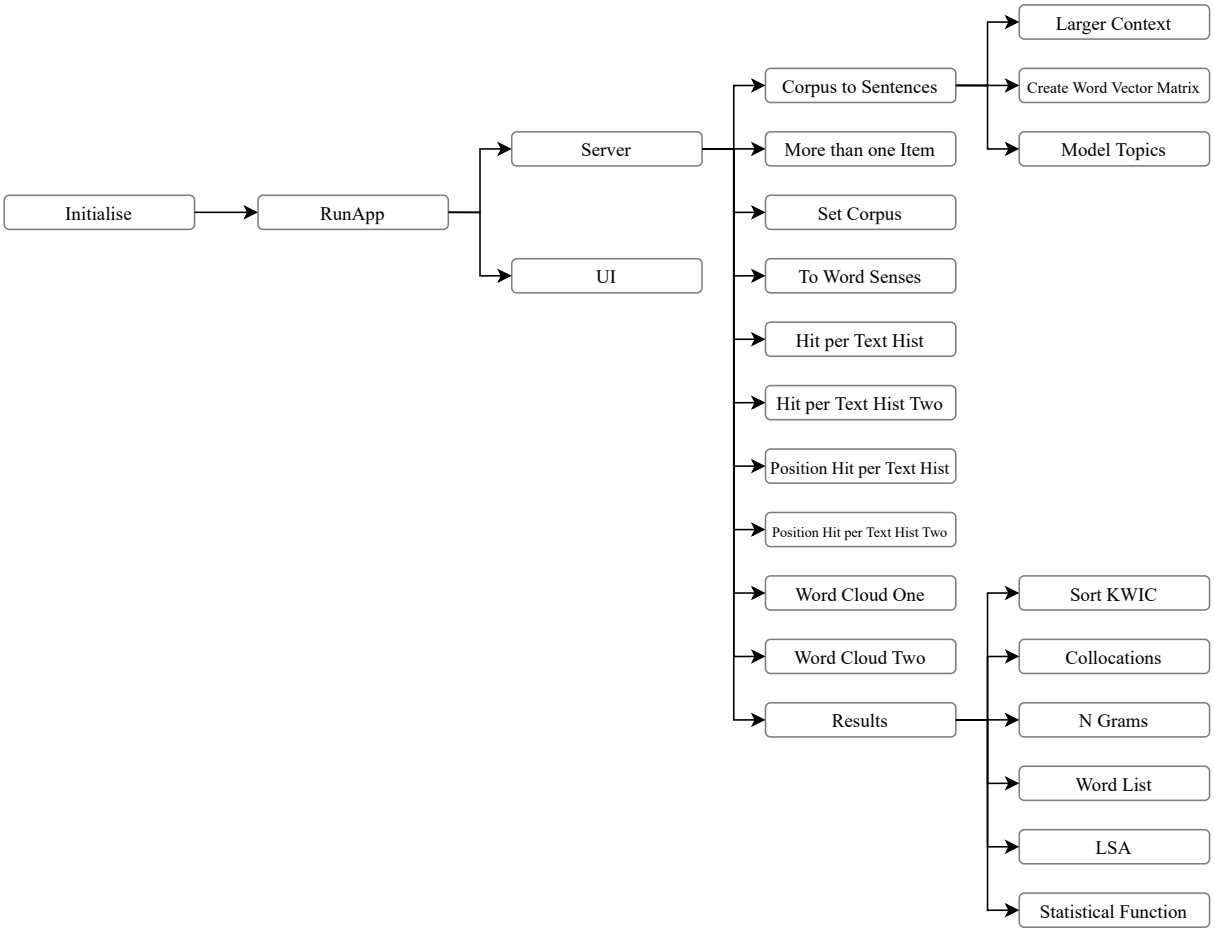


Abbildung 5.13: Schematische Darstellung der Funktions- und Datenaufrufe in HanConc

gabe von Pflichtexemplaren jeder Dissertation an die Universitätsbibliothek vor. Die Abgabe kann analog oder digital erfolgen. Bei den hier verwendeten Texten handelt es sich um die digital eingereichten Exemplare.

Die Texte sind zusammen mit einer Tabelle mit Metadaten wie dem Titel, Autor_in und Erscheinungsjahr zur Verfügung gestellt worden. Alle Texte sind im PDF Format gespeichert. Aufgrund des Schriftbildes ist zu vermuten, dass ein Großteil aus Word exportiert oder mit LaTeX gesetzt wurde.

Die Aufbereitung der Texte erfolgt in mehreren Schritten[42], welche in Abbildung 5.14 skizziert werden. Handelt es sich bei den Ursprungsdateien um andere Dateiformate als PDF, entfallen gegebenenfalls einige Aufbereitungsschritte. Kapitel 5.3.3.1 beschreibt die Umwandlung von PDF zu TXT Dateien und die gleichzeitige Bereinigung derer um Fragmente wie Zahlen, Überschriften und Verweise. Die gesäuberten Texte werden anschließend linguistisch annotiert (siehe Kapitel 5.3.3.2). Erst nach diesem Schritt sind die Texte in einzelne Sätze unterteilt und mit Tags versehen, welche die Wortart oder andere Metainformationen beinhalten. In einem letzten Aufbereitungsschritt werden die annotierten Texte in HanConc integriert (siehe Kapitel 5.3.3.3).

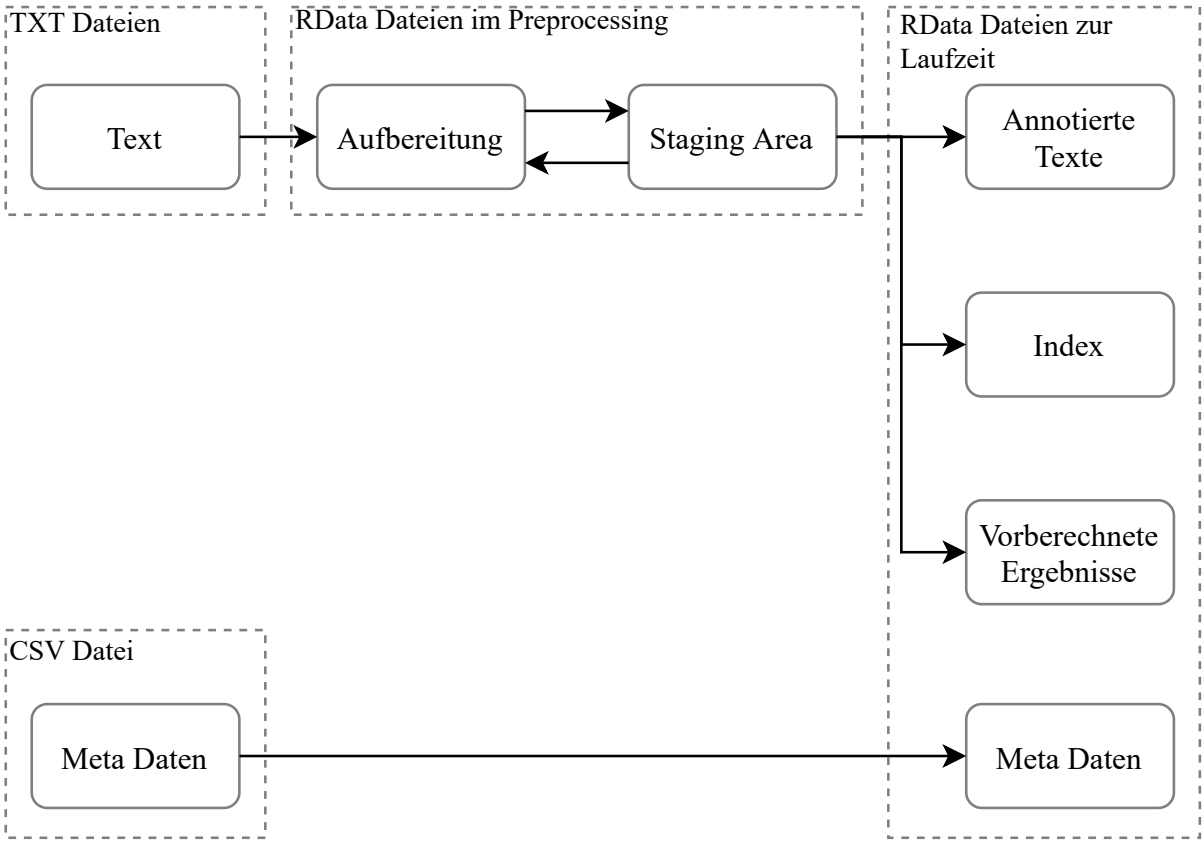


Abbildung 5.14: Architektur der Textaufbereitung

[42]Die Textaufbereitungsschritte können analog auch für jede andere Textgrundlage verwendet werden, um eigene Texte für HanConc aufzubereiten.

Tabelle 5.4: Reguläre Ausdrücke (RegEx) zum Entfernen von Textfragmenten

Fragment	Beispiel	Regulärer Ausdruck	Ersetzung
Zahlen	15	$\backslash s[0-9]^*\backslash s$	Integer
Gleitkommazahlen	1,3	$\backslash s[0-9\backslash.]^*, [0-9]^*$	Float
Prozentwerte	15,29 %	$\backslash s[0-9\backslash s\backslash, \backslash]^*\%$	Percentage
Verweis (IEEE)	[12]	$\backslash[[0-9]^*\backslash]$	Quote
Verweis (DIN 1505-3)	(Mai15)	$([A-Za-z]\{3\}[0-9]\{2\})$	Quote
Verweis (DIN 1505-3)	[Mai15]	$\backslash[[A-Za-z]\{3\}[0-9]\{2\}\backslash]$	Quote
Verweis (MLA)	(Maier 2015, 198)	$([A-Za-z]\backslash s[0-9]\{4\}, [0-9]^*)$	Quote
Verweis (MLA)	(Maier 2015)	$([A-Za-z]\backslash s[0-9]\{4\})$	Quote

5.3.3.1 Umwandlung von PDF zu TXT Dateien

Da das PDF Format darauf ausgelegt ist, nicht verändert zu werden, müssen die Dissertationen für die Weiterverarbeitung zunächst mit Hilfe des Standardlinuxwerkzeugs pdftotext in TXT Dateien konvertiert werden. Aufgrund der unterschiedlich formatierten PDF Dateien und weil die Metainformationen nicht einheitlich ausgelesen werden können, enthalten die TXT Dateien einen hohen Anteil an Textfragmenten wie Überschriften, Bildunterschriften, Formeln oder Zahlen. Diese Fragmente müssen in mehreren Aufbereitungsschritten entfernt werden, da sie keinen zielführenden Informationsgehalt besitzen (Kapitelüberschriften), als Plagiat genutzt werden können (Zahlen), Fehler beim Taggen verursachen können (Sonderzeichen) oder vom Frontend nicht wiedergegeben werden können (Formeln).

Tabelle 5.4 zeigt eine Auswahl an regulären Ausdrücken (RegEx), die verwendet werden können, um die oben genannten Fragmente zu entfernen.

Nach jedem Bearbeitungsschritt werden die Texte zwischengespeichert, um später die Möglichkeit zu haben, etwa Korpora bestehend aus Bildunterschriften oder Tabellenüberschriften zu generieren.

Die bereinigten Texte müssen noch weiterverarbeitet werden, um Annotationen hinzuzufügen, die Suche durch einen Index zu beschleunigen, verschiedene Ergebnisse vorzuberechnen und dadurch Computerlaufzeiten zu sparen.

5.3.3.2 Annotierung von TXT Dateien zu XML Dokumenten

Für die Suche mit HanConc müssen den Texten drei Informationsarten hinzugefügt werden: Satzgrenzen, Lemmata und Wortarten. Der Bereich um einen Suchbegriff kann mittels einer Begrenzung der Wörter oder Zeichen zur Rechten und Linken (vgl. AntConc oder BNC) oder mittels der Satzgrenzen limitiert werden. Die Satzgrenzen sind in HanConc definiert als fortlaufende Begrenzung durch eines der folgenden Zeichen: „ . ”, „ ! ” oder „ ? ”. Satzgrenzen wurden als Begrenzung der KWIC gewählt, um sicherzugehen, dass komplexere Konstruktionen wie zum Beispiel Partizipialkonstruktionen komplett wiedergegeben werden und ein alleiniges Anzeigen von Teilen der Konstruktionen aufgrund eines überlangen Satzes verhindert wird.

Eine eigenständige Software zu entwickeln, die Texte in Sätze aufteilt und annotiert, würde

ein eigenes Forschungsprojekt darstellen, das den Rahmen und die Möglichkeiten dieser Arbeit sprengen würde. Daher wurde hierfür ein externes Softwarepaket ausgewählt, das folgende Anforderungen erfüllt:

- Kostenlos
- Plattformunabhängig
- Möglichst einfach zu bedienen oder aus R heraus aufrufbar
- Verfügbar zumindest für englische, deutsche, spanische und französische Texte
- Sentence Splitter
- PoS Tagger
- Auf eigenen Geräten einsetzbar
- Rückgabe der Texte in strukturierter Form, d.h. als XML oder JSON Datei

Diese Anforderungen sollen sicherstellen, dass das Taggen eine möglichst niedrige Hürde darstellt und Anpassungen, wie etwa die Implementierung einer neuen Sprache, möglichst einfach realisiert werden können. In die engere Auswahl sind der Stanford Part-of-Speech (PoS) (Manning & Schutze 1999) und der Stuttgarter TreeTagger (Schmid 2013) gekommen. Beide sind, im Gegensatz zu etwa Googles Tagger[43], der nur über eine online API verfügbar ist und das Hochladen der Texte in die Google Cloud voraussetzt, lokal auf verschiedenen Betriebssystemen installierbar. Außerdem stellen der Stanford PoS Tagger und der Stuttgarter TreeTagger Erweiterungen für diverse Sprachen zur Verfügung und sind direkt aus R aufrufbar.

Der Stuttgarter TreeTagger kann über das koRpus (Michalke 2018) Paket aufgerufen werden. Der Stanford PoS Tagger ist in Java geschrieben und kann mit wenigen Parametern über die Kommandozeile gestartet werden. Hierdurch ist es möglich, einen R Wrapper zu schreiben, welcher die Parameter aus R heraus an die Kommandozeile übergibt.

Für die Textaufbereitung für HanConc wurde der Stuttgarter TreeTagger nicht ausgewählt, da er sich laut Aussage auf der dazugehörigen Webseite[44] und nach der Erfahrung des Autors nur schwerlich unter Windows installieren lässt.

Der Stanford PoS Tagger gibt wahlweise TXT oder XML Dateien zurück. In jedem Fall müssen die Texte daraufhin geparset werden, d.h. die Tagstruktur der TXT oder XML Dateien muss in ein für R lesbares Format überführt werden. Die folgenden Quellcodes 5.1 bis 5.4 zeigen den gleichen Satz als Rohtext, getagt als XML und in der R Struktur:

```
Obwohl "die Grammatik" allgemein als die feste Grundstruktur einer Sprache
    gilt – ihr Skelett sozusagen –, ist sie dennoch sprachlichem Wandel
    unterworfen.
```

Quellcode 5.1: Satzbeispiel als Rohtext

[43] Verfügbar unter https://cloud.google.com/natural-language (Stand: 10. März 2020)
[44] https://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/ (Stand: 10. März 2020)

Der obige Satz soll nun mit Software aus dem Stanford NLP Paket annotiert werden. Hierfür wird das Core Paket genutzt, mit welchem mehrere Annotationsschichten auf den Text gelegt werden können. In HanConc sind bisher drei Schichten (Wort, Lemma, PoS-Tag) vorgesehen. Denkbar ist jedoch, einzelne Schichten im Verlauf der Annotation zum Beispiel durch Named Entities zu ersetzen.

Quellcode 5.2 zeigt den Aufruf der Java Applikation des Stanford Core NLP Pakets aus der Kommandozeile. Es wird bewusst darauf verzichtet, einen Java Wrapper[45] zu schreiben, um eine reine R Lösung zu ermöglichen. Stattdessen wird Java über die Kommandozeile aus R aufgerufen:

```
java -cp "*" -Xmx2g edu.stanford.nlp.pipeline.StanfordCoreNLP -annotators
    tokenize,ssplit,pos,lemma -props StanfordCoreNLP-german.properties -file
     sample_input.txt
```

Quellcode 5.2: Stanford Core NLP Programmaufruf

Der Output aus Quellcode 5.2 wird als XML Quellcode in 5.3 gezeigt. Jede Annotationsschicht wird als eigenes Tag realisiert. In diesem Fall handelt es sich um: Root, Document, Sentences, Sentence, Tokens, Token, Word, Lemma, CharacterOffsetBegin, CharacterOffsetEnd und POS. Abbildung 5.15 zeigt die Struktur der XML Dokumente, welche das Stanford Core NLP Paket als Output generiert. Das tatsächliche XML Dokument wird in Quellcode 5.3 gezeigt.

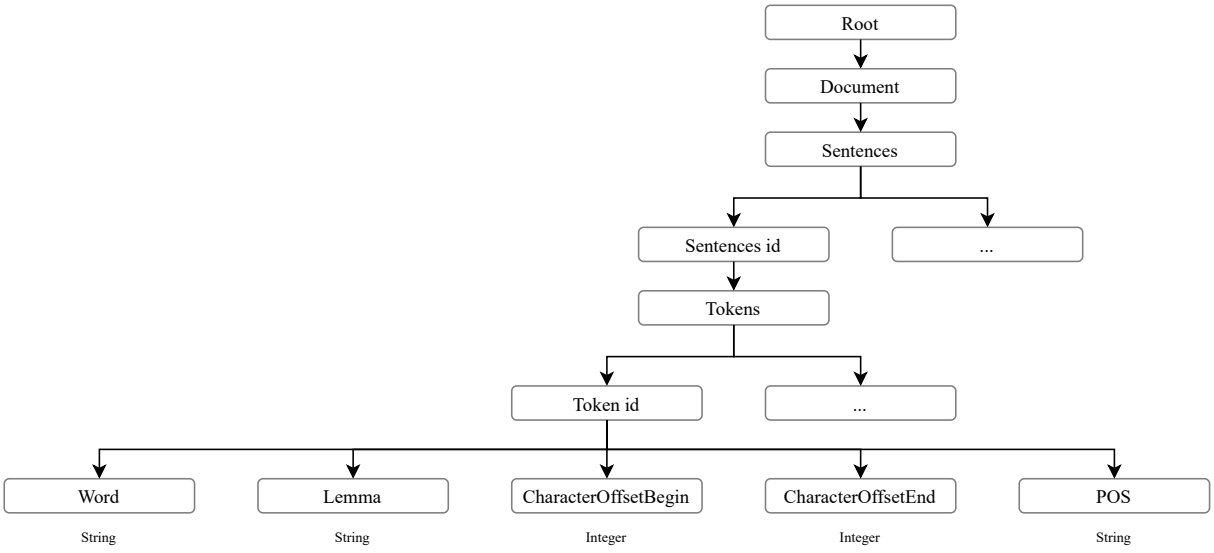


Abbildung 5.15: XML Dokumentstruktur zu Quellcode 5.3

Jedes Dokument ist wie folgt organisiert: Unter einem „Root" Knoten befindet sich das eigentliche Dokument, welches wiederum aus einzelnen Sätzen besteht. Der Sentencesplitter, welcher die Satzgrenzen definiert, ist durch das „Sentence" Tag realisiert. Jeder Satz besteht aus „Tokens". In diesem Fall ist jedes Wort ein eigenes Token. Entsprechend der Konfiguration aus Quellcode 5.2 ist jedes Token mit fünf Tags versehen: mit dem ursprünglichen Wort als String,

[45] https://stanfordnlp.github.io/CoreNLP/api.html (Stand: 10. März 2020)

dem Lemma[46] als String, der Position des ersten Buchstabens des Wortes von eins an gezählt und exklusiv der Leerzeichen als zum String konvertierter Integer, mit selbigem für den letzten Buchstaben des Wortes und dem PoS Tag als String. Hier wird das Stuttgart-Tübingen-Tagset (STTS) verwendet (Schiller, Teufel, Thielen & Stöckert 1999).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
  <document>
    <sentences>
      <sentence id="1">
        <tokens>
          <token id="1">
            <word>Obwohl</word>
            <lemma>obwohl</lemma>
            <CharacterOffsetBegin>1</CharacterOffsetBegin>
            <CharacterOffsetEnd>7</CharacterOffsetEnd>
            <POS>KOUS</POS>
          </token>
          <token id="2">
            <word>``</word>
            <lemma>``</lemma>
            <CharacterOffsetBegin>8</CharacterOffsetBegin>
            <CharacterOffsetEnd>9</CharacterOffsetEnd>
            <POS>CARD</POS>
          </token>
          <token id="3">
            <word>die</word>
            <lemma>die</lemma>
            <CharacterOffsetBegin>9</CharacterOffsetBegin>
            <CharacterOffsetEnd>12</CharacterOffsetEnd>
            <POS>ART</POS>
          </token>
          <token id="4">
            <word>Grammatik</word>
            <lemma>grammatik</lemma>
            <CharacterOffsetBegin>13</CharacterOffsetBegin>
            <CharacterOffsetEnd>22</CharacterOffsetEnd>
            <POS>NN</POS>
          </token>
          <token id="5">
            <word>``</word>
            <lemma>``</lemma>
            <CharacterOffsetBegin>22</CharacterOffsetBegin>
            <CharacterOffsetEnd>23</CharacterOffsetEnd>
            <POS>CARD</POS>
          </token>
```

[46] Für das Deutsche ist leider kein Lemmatisierer verfügbar. Das hier verwendete Tag enthält nur das kleingeschriebene Wort.

```
<token id="6">
  <word>allgemein</word>
  <lemma>allgemein</lemma>
  <CharacterOffsetBegin>24</CharacterOffsetBegin>
  <CharacterOffsetEnd>33</CharacterOffsetEnd>
  <POS>ADJD</POS>
</token>
<token id="7">
  <word>als</word>
  <lemma>als</lemma>
  <CharacterOffsetBegin>34</CharacterOffsetBegin>
  <CharacterOffsetEnd>37</CharacterOffsetEnd>
  <POS>KOKOM</POS>
</token>
<token id="8">
  <word>die</word>
  <lemma>die</lemma>
  <CharacterOffsetBegin>38</CharacterOffsetBegin>
  <CharacterOffsetEnd>41</CharacterOffsetEnd>
  <POS>ART</POS>
</token>
<token id="9">
  <word>feste</word>
  <lemma>feste</lemma>
  <CharacterOffsetBegin>42</CharacterOffsetBegin>
  <CharacterOffsetEnd>47</CharacterOffsetEnd>
  <POS>ADJA</POS>
</token>
<token id="10">
  <word>Grundstruktur</word>
  <lemma>grundstruktur</lemma>
  <CharacterOffsetBegin>48</CharacterOffsetBegin>
  <CharacterOffsetEnd>61</CharacterOffsetEnd>
  <POS>NN</POS>
</token>
<token id="11">
  <word>einer</word>
  <lemma>einer</lemma>
  <CharacterOffsetBegin>62</CharacterOffsetBegin>
  <CharacterOffsetEnd>67</CharacterOffsetEnd>
  <POS>ART</POS>
</token>
<token id="12">
  <word>Sprache</word>
  <lemma>sprache</lemma>
  <CharacterOffsetBegin>68</CharacterOffsetBegin>
  <CharacterOffsetEnd>75</CharacterOffsetEnd>
  <POS>NN</POS>
</token>
```

```
92          <token id="13">
93            <word>gilt</word>
94            <lemma>gilt</lemma>
95            <CharacterOffsetBegin>76</CharacterOffsetBegin>
96            <CharacterOffsetEnd>80</CharacterOffsetEnd>
97            <POS>VVFIN</POS>
98          </token>
99          <token id="14">
100           <word>–</word>
101           <lemma>–</lemma>
102           <CharacterOffsetBegin>82</CharacterOffsetBegin>
103           <CharacterOffsetEnd>83</CharacterOffsetEnd>
104           <POS>$[</POS>
105         </token>
106         <token id="15">
107           <word>ihr</word>
108           <lemma>ihr</lemma>
109           <CharacterOffsetBegin>84</CharacterOffsetBegin>
110           <CharacterOffsetEnd>87</CharacterOffsetEnd>
111           <POS>PPOSAT</POS>
112         </token>
113         <token id="16">
114           <word>Skelett</word>
115           <lemma>skelett</lemma>
116           <CharacterOffsetBegin>88</CharacterOffsetBegin>
117           <CharacterOffsetEnd>95</CharacterOffsetEnd>
118           <POS>NN</POS>
119         </token>
120         <token id="17">
121           <word>sozusagen</word>
122           <lemma>sozusagen</lemma>
123           <CharacterOffsetBegin>96</CharacterOffsetBegin>
124           <CharacterOffsetEnd>105</CharacterOffsetEnd>
125           <POS>ADV</POS>
126         </token>
127         <token id="18">
128           <word>––</word>
129           <lemma>––</lemma>
130           <CharacterOffsetBegin>106</CharacterOffsetBegin>
131           <CharacterOffsetEnd>107</CharacterOffsetEnd>
132           <POS>APPRART</POS>
133         </token>
134         <token id="19">
135           <word>,</word>
136           <lemma>,</lemma>
137           <CharacterOffsetBegin>107</CharacterOffsetBegin>
138           <CharacterOffsetEnd>108</CharacterOffsetEnd>
139           <POS>$,</POS>
140         </token>
```

```
<token id="20">
  <word>ist</word>
  <lemma>ist</lemma>
  <CharacterOffsetBegin>109</CharacterOffsetBegin>
  <CharacterOffsetEnd>112</CharacterOffsetEnd>
  <POS>VAFIN</POS>
</token>
<token id="21">
  <word>sie</word>
  <lemma>sie</lemma>
  <CharacterOffsetBegin>113</CharacterOffsetBegin>
  <CharacterOffsetEnd>116</CharacterOffsetEnd>
  <POS>PPER</POS>
</token>
<token id="22">
  <word>dennoch</word>
  <lemma>dennoch</lemma>
  <CharacterOffsetBegin>117</CharacterOffsetBegin>
  <CharacterOffsetEnd>124</CharacterOffsetEnd>
  <POS>ADV</POS>
</token>
<token id="23">
  <word>sprachlichem</word>
  <lemma>sprachlichem</lemma>
  <CharacterOffsetBegin>125</CharacterOffsetBegin>
  <CharacterOffsetEnd>137</CharacterOffsetEnd>
  <POS>ADJA</POS>
</token>
<token id="24">
  <word>Wandel</word>
  <lemma>wandel</lemma>
  <CharacterOffsetBegin>138</CharacterOffsetBegin>
  <CharacterOffsetEnd>144</CharacterOffsetEnd>
  <POS>NN</POS>
</token>
<token id="25">
  <word>unterworfen</word>
  <lemma>unterworfen</lemma>
  <CharacterOffsetBegin>145</CharacterOffsetBegin>
  <CharacterOffsetEnd>156</CharacterOffsetEnd>
  <POS>VVPP</POS>
</token>
<token id="26">
  <word>.</word>
  <lemma>.</lemma>
  <CharacterOffsetBegin>156</CharacterOffsetBegin>
  <CharacterOffsetEnd>157</CharacterOffsetEnd>
  <POS>$.</POS>
</token>
```

```
190          </tokens>
191        </sentence>
192      </sentences>
193    </document>
194 </root>
```

Quellcode 5.3: Satzbeispiel als getagtes XML Dokument

5.3.3.3 Integration von geparseten XML Dateien in R

Im Sinne einer kompletten Datenhaltung in R muss das oben beschriebene XML Dokument in ein entsprechendes Format geparset werden. Um die hierarchische Struktur beizubehalten, wird eine dreistufige R Liste gewählt. Bei Listen handelt es sich um R Objekte, die wiederum andere Objekte enthalten (R Core Team 2021). In diesem Fall handelt es sich um eine Liste, welche das Gesamtkorpus darstellt. In ihr befinden sich weitere Listen, welche jeweils einen Text repräsentieren. Jeder Text wiederum besteht aus je einer Tabelle mit drei Spalten für Wörter, Lemmata und PoS Tags pro Satz. Abbildung 5.16 skizziert diesen Aufbau inklusive der enthaltenen R Datenstrukturen.

Quellcode 5.4 zeigt den Aufbau des abschließenden DataFrames am vorangegangenen Beispiel aus Quellcode 5.1. Es besteht aus drei Spalten: Wort, Lemma und PoS Tag[47]. Jede Zeile entspricht einem Wort aus dem Ursprungstext. Da die Spalten benannt sind und entsprechend aufgerufen werden, kann eine Veränderung der Datenstruktur jederzeit vorgenommen werden, ohne Teile des Suchalgorithmus' verändern zu müssen.

```
1  Wort          | Lemma          | POS-Tag
2  Obwohl        | obwohl         | KOUS
3  ``            | ``             | CARD
4  die           | die            | ART
5  Grammatik     | grammatik      | NN
6  ``            | ``             | CARD
7  allgemein     | allgemein      | ADJD
8  als           | als            | KOKOM
9  die           | die            | ART
10 feste         | feste          | ADJA
11 Grundstruktur | grundstruktur  | NN
12 einer         | einer          | ART
13 Sprache       | sprache        | NN
14 gilt          | gilt           | VVFIN
15 -             | -              | APPRART
16 ihr           | ihr            | PPOSAT
17 Skelett       | skelett        | NN
18 sozusagen     | sozusagen      | ADV
19 -             | -              | APPRART
```

[47] Hierbei handelt es sich um die Standardstruktur von HanConc. Sollen andere Annotationsschichten wie etwa eine NER hinzugefügt werden, so können diese entweder das DataFrame um eine weitere Spalte erweitern oder eine der bestehenden Spalten ersetzen. Es wird empfohlen, die zweite Spalte zu ersetzen, da diese im Moment nur das jeweilige Wort mit Kleinbuchstaben enthält.

```
20 ,              | ,              | $ ,
21 ist            | ist            | VAFIN
22 sie            | sie            | PPER
23 dennoch        | dennoch        | ADV
24 sprachlichem   | sprachlichem   | ADJA
25 Wandel         | wandel         | NN
26 unterworfen    | unterworfen    | VVPP
27 .              | .              | $ .
```

Quellcode 5.4: Satzbeispiel mit Lemma und POS Tag je Wort als R Datenstruktur

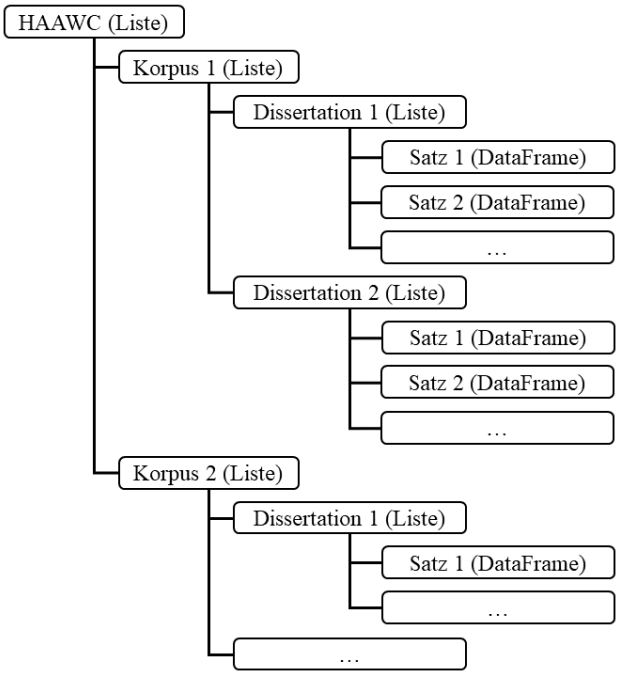


Abbildung 5.16: Organisationsstruktur der Korpora mit entsprechender R Datenstruktur

Durch den Aufbau in R Listen und DataFrames ist es möglich, unabhängig von HanConcs Weboberfläche, quantitative linguistische Forschung zu betreiben, ohne die Programmiersprache zu wechseln. Außerdem ist HanConc so programmiert, dass auch Programmierneulinge den Code verstehen und manipulieren können. Schreibberater_innen und Linguist_innen sollen mit einem Grundverständnis von *for* Schleifen, *if/else* Bedingungen, R DataFrames und Listen die Möglichkeit haben, mit HanConc selbständig neue Ergebnisse zu generieren und Untersuchungen durchzuführen.

5.3.4 Ergänzung von HanConc um eigene Zusatzfunktionen

Die offene und einsteigerfreundliche Programmierung von HanConc soll an zwei Beispielen demonstriert werden. Es soll zunächst überprüft werden, ob Maschinenbauingenieur_innen längere Adjektive benutzen als Elektrotechnikingenieur_innen. Die in Quellcode 5.5 aufgezeigte Lösung ist, bis auf die dreifach geschachtelte Schleife, sehr einfach gehalten und mit weniger als 50 Zeilen deutlich kürzer als eine Alternative in X-Path oder X-Query.

```
# analysis of mean word length of adjectives in felt
################################################

wortLaengeElt = 0
index = 1

for(dokument in 1:length(felt)){
  for(satz in 1:length(felt[[dokument]])){
    for(wort in 1:length(felt[[dokument]][[satz]])){
      if(felt[[dokument]][[satz]]$pos[wort] == "ADJD"){
      wortLaengeElt[index] = nchar(felt[[dokument]][[satz]]$pos[wort])
      index = index + 1
      }
    }
  }
}

# print mean value and standard deviation
mean(wortLaengeElt);sd(wortLaengeElt)

# analysis of mean word length of adjectives in fmas
################################################

wortLaengeMas = 0
index = 1

for(dokument in 1:length(fmas)){
  for(satz in 1:length(fmas[[dokument]])){
    for(wort in 1:length(fmas[[dokument]][[satz]])){
      if(fmas[[dokument]][[satz]]$pos[wort] == "ADJD"){
      wortLaengeMas[index] = nchar(fmas[[dokument]][[satz]]$pos[wort])
      index = index + 1
      }
    }
  }
}

# print mean value and standard deviation
mean(wortLaengeMas);sd(wortLaengeMas)

# print results of t-test
t.test(wortLaengeMas,wortLaengeElt)
```

Quellcode 5.5: Beispielcode in R zur Berechnung und Analyse der durschnittlichen Wortlänge von Adjektiven bei Maschinenbauingenieur_innen und Elektrotechnikingenieur_innen

In einem zweiten Beispiel soll das grammatikalische Geschlecht eines eingegebenen Substantivs bestimmt werden. Als Grundlage dafür dienen die dem Substantiv vorausgehenden

Artikel. Es ist zu bedenken, dass dieses Vorgehen nur bei Substantiven im Singular funktioniert (Kunkel-Razum 2006, 152-155). Es soll also erneut über die R Liste iteriert werden (siehe Kapitel 5.3.3.3) und, sobald das entsprechende Wort gefunden wurde, nach Artikeln in unmittelbarer Nähe auf der linken Seite gesucht werden. Außerdem sollen eventuelle Adjektive berücksichtigt werden, die direkt vor dem Substantiv stehen können. Als grammatikalische Referenz dient Tabelle 5.5. Sie enthält alle bestimmten und unbestimmten Artikel des Deutschen (Kunkel-Razum 2006, 292)[48].

Tabelle 5.5: Bestimmte und unbestimmte Artikel des Deutschen nach Kasus und Genus

	maskulin	feminin	neutral	Plural
Nominativ	der	die	das	die
Genitiv	des	der	des	der
Dativ	dem	der	dem	den
Akkusativ	den	die	das	die
Nominativ	ein	eine	ein	
Genitiv	eines	einer	eines	
Dativ	einem	einer	einem	
Akkusativ	einen	eine	ein	

Um das grammatikalische Geschlecht zu bestimmen, sollen die verwendeten Artikel ausgezählt werden. Mehrfach vorkommende Artikel werden gewichtet, sodass ein Vorkommen von „der" zu einem Drittel für maskulin und zu zwei Dritteln zu feminin gezählt wird[49]. Daraus ergibt sich die Erweiterung von Tabelle 5.5 um eine Gewichtung zu Tabelle 5.6. Mit „den", „die", „das" und „einen" gibt es nur vier aus 24 Artikeln, die exklusiv geschlechtsspezifisch sind. Alle übrigen Artikel werden geschlechtsübergreifend verwendet. Im Fall von dreifach verwendeten Artikeln wie „der" ergibt sich aufgrund der Gewichtung keine Tendenz zu einem bestimmten Geschlecht. Die Annahme ist, dass bei einer genügend großen Anzahl an Beobachtungen diese Tendenz überlagert wird.

Wie im obigen Beispiel zur Untersuchung der Wortlänge von Adjektiven soll erneut über alle Sätze iteriert werden. Wird das entsprechende Suchwort gefunden, wird die Position davor auf einen Artikel überprüft. Wird ein Artikel gefunden, wird dieser auf sein Geschlecht überprüft und der Zähler des entsprechenden Geschlechts um den jeweiligen Wert, siehe Tabelle 5.6, erhöht. Befindet sich auf der Position vor dem Substantiv ein Adjektiv, so wird eine weitere Position davor auf einen Artikel überprüft.

Mit dem Wort „Versuch" soll ein Beispiel konstruiert werden, bei dem auf RegEx bewusst verzichtet wird, da sie den Quellcode unnötig verkomplizieren[50]. Es ergibt sich folgender Quell-

[48] Demonstrativpronomen wurden aus Gründen der Lesbarkeit nicht mit aufgenommen. Bei einer tatsächlichen Untersuchung müssten sie analog berücksichtigt werden.

[49] „der" kann entweder Nominativ Singular maskulin, Genitiv Singular feminin oder Dativ Singular feminin. Daher ist die Wahrscheinlichkeit doppelt so hoch, dass ein Substantiv, das auf ein „der" folgt, feminin und nicht maskulin ist.

[50] Mittels RegEx kann die Trefferanzahl erhöht werden. Dies beeinträchtigt jedoch die Genauigkeit. Wird das Wort „Versuch" zu „Versuch[a-z]{2}" um zwei nicht definierte Buchstaben erweitert, so muss eine zusätzliche

Tabelle 5.6: Statistische Gewichtung bestimmter und unbestimmter Artikel des Deutschen nach Kasus und Genus

	maskulin	feminin	neutral
Nominativ	der (1/3)	die (1/2)	das (1/2)
Genitiv	des (1/2)	der (1/3)	des (1/2)
Dativ	dem (1/2)	der (1/3)	dem (1/2)
Akkusativ	den	die (1/2)	das (1/2)
Nominativ	ein (1/3)	eine (1/2)	ein (1/3)
Genitiv	eines (1/2)	einer (1/2)	eines (1/2)
Dativ	einem (1/2)	einer (1/2)	einem (1/2)
Akkusativ	einen	eine (1/2)	ein (1/3)

code[51]:

```
################################################
# create gender variables
################################################
male = 0
female = 0
neuter = 0

################################################
# main loop
################################################

corpus = fmas #could be any other corpus

for(dokument in 1:length(corpus)){
  for(satz in 1:length(corpus[[dokument]])){
    for(wort in 1:length(corpus[[dokument]][[satz]])){
      # do nothing if it is the first word of the sentence
      if(wort != 1){
        # check for the word
        if(corpus[[dokument]][[satz]]$word[wort] == "Versuch"){
          # check for previous word only if it is not the second word
          if(wort != 2){
            prevWord = corpus[[dokument]][[satz]]$pos[wort - 1]
            if(prevWord == "JJ" || prevWord == "JR" || prevWord == "JS"){
              if(corpus[[dokument]][[satz]]$pos[wort - 2] == "DT"){
                # check for actual gender
                if(corpus[[dokument]][[satz]]$word[wort - 2] == "der"){
                  male = male   + (1/3)
                  female = female + (2/3)
```

Bedingung in den Quellcode eingefügt werden, um Pluralformen auszuschließen. Außerdem kann es bei anderen Beispielen dazu kommen, dass nicht beabsichtigte Wörter zurückgegeben werden. Eine Erweiterung von „Tau" im Sinne des griechischen Buchstabens um zwei Buchstaben kann zum Beispiel auch den Vogel „Taube" betreffen und zu falschen Ergebnissen führen.

[51] Unbestimmte Artikel wurden nicht berücksichtigt, um die Lesbarkeit des Codes zu gewährleisten.

```
30                    neuter = neuter
31                  }
32                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "des"){
33                    male   = male     + (1/2)
34                    female   = female
35                    neuter   = neuter   + (1/2)
36                  }
37                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "dem"){
38                    male   = male     + (1/2)
39                    female   = female
40                    neuter   = neuter   + (1/2)
41                  }
42                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "den"){
43                    male   = male     + 1
44                    female   = female
45                    neuter   = neuter
46                  }
47                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "die"){
48                    male   = male
49                    female   = female   + 1
50                    neuter   = neuter
51                  }
52                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "das"){
53                    male   = male
54                    female   = female
55                    neuter   = neuter   + (1/1)
56                  }
57                }
58              }else{
59                # without preceeding adjective
60                if(corpus[[dokument]][[satz]]$pos[wort - 2] == "DT"){
61                  # check for actual gender
62                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "der"){
63                    male = male    + (1/3)
64                    female = female + (2/3)
65                    neuter = neuter
66                  }
67                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "des"){
68                    male   = male     + (1/2)
69                    female   = female
70                    neuter   = neuter   + (1/2)
71                  }
72                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "dem"){
73                    male   = male     + (1/2)
74                    female   = female
75                    neuter   = neuter   + (1/2)
76                  }
77                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "den"){
78                    male   = male     + 1
```

```
                    female  = female
                    neuter  = neuter
                  }
                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "die"){
                    male  = male
                    female  = female  + 1
                    neuter  = neuter
                  }
                  if(corpus[[dokument]][[satz]]$word[wort - 2] == "das"){
                    male  = male
                    female  = female
                    neuter  = neuter  + (1/1)
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Quellcode 5.6: Beispielcode in R zur Bestimmung des grammatikalischen Geschlechts

Das Geschlecht ergibt sich aus dem Vergleich der Variablen „male", „female" und „neuter". Die Variable mit dem höchsten Wert entspricht dem grammatikalischen Geschlecht.

Dieser Quellcode kann in HanConc eingefügt werden, um Probleme Studierender in Bezug auf die korrekte Verwendung von Artikeln zu adressieren. Eine Darstellung der Auswertung im Frontend von HanConc könnte auf zwei Arten erfolgen: Eine einfache Darstellung könnte ein Symbol (etwa ♂, ♀ oder o) für das wahrscheinlichste Geschlecht nutzen oder für alle drei Geschlechter jeweils die Wahrscheinlichkeit angeben. Während die erste Möglichkeit vereinfachend wirkt, bringt die zweite Lösung die Ambiguität von Sprache zum Ausdruck. In beiden Fällen sollte jedoch die technische Umsetzung mittels eines zusammengesetzten HTML Elements geschehen. Eine solche Herangehensweise hat den Vorteil, dass grundlegende HTML und R Kenntnisse ausreichend sind.

Im nun folgenden Abschnitt werden alle Schritte skizziert, die notwendig sind, um die oben beschriebene Funktion in ein R Shiny Frontend wie etwa dem von HanConc zu integrieren. Als funktionsfähiges Minimalbeispiel ist eine Applikation ausreichend, die nur aus einer $ui.R$ und einer $server.R$ Datei besteht. Die $server.R$ Datei enthält eine Funktion, die basierend auf den Nutzer_inneneingaben eine Ergebnisliste generiert. Diese Liste soll in diesem Beispiel $results$ heißen und unter anderem die Elemente $male$, $female$ und $neuter$ aus Quellcode 5.6 enthalten, welche den Datentyp Integer haben. Die Aufbereitung wird folgendermaßen vorgenommen:

- Berechnung der Summe aller gewerteten Substantive
- Berechnung der relativen Häufigkeiten

- Formatierung zu Prozentwerten, welche auf die zweite Nachkommastelle gerundet sind
- Gegebenenfalls eine Umformatierung in das deutsche Zahlenformat
- Integration in eine HTML Tabelle
- Integration in das Frontend ($ui.R$)

Die ersten vier Punkte können in wenigen Zeilen R Code programmiert werden. Quellcode 5.7 formatiert die absoluten Zahlen aus Quellcode 5.6 in das gewünschte Format. Es ist zu beachten, dass der Quellcode dahingehend geschrieben wurde, möglichst einsteigerfreundlich zu sein. Aus Gründen der Übersicht und Lesbarkeit dieser Arbeit wird das Beispiel stark verkürzt dargestellt.

```
# Number of found nouns
numberOfNouns = results$male + results$female + results$neuter

# Calculate shares in percent
shareMale = results$male / numberOfNouns * 100
shareFemale = results$female / numberOfNouns * 100
shareNeuter = results$neuter / numberOfNouns * 100

# Round shares to 2nd digit
shareMaleRounded = round(shareMale, 2)
shareFemaleRounded = round(shareFemale, 2)
shareNeuterRounded = round(shareNeuter, 2)

# Format as string
shareMaleRoundedString = toString(shareMaleRounded)
shareFemaleRoundedString = toString(shareFemaleRounded)
shareNeuterRoundedString = toString(shareNeuterRounded)

# Convert to German number style
# ("." has to be escaped with \\;
# otherwise it is taken as RegEx for any character)
shareMaleGermanString = gsub("\\.", "," , shareMaleRounded)
shareFemaleGermanString = gsub("\\.", "," , shareFemaleRounded)
shareNeuterGermanString = gsub("\\.", "," , shareNeuterRounded)

# Append % sign
shareMaleFinal = paste0(shareMaleGermanString, " %")
shareFemaleFinal = paste0(shareFemaleGermanString, " %")
shareNeuterFinal = paste0(shareNeuterGermanString, " %")
```

Quellcode 5.7: Beispielcode in R zur Aufarbeitung der Ergebnisse aus Quellcode 5.6

Die Gestaltung des Frontends kann über CSS vorgenommen werden. An dieser Stelle soll allerdings nur eine Einbettung als einfache HTML Tabelle vorgenommen werden. Der folgende Quellcode 5.8 beschreibt die Integration der Ergebnisse aus Quellcode 5.7 in das Frontend.

Damit Einsteiger_innen unter Zuhilfenahme kurzer Anleitungen selbst den Code verstehen und manipulieren können, liegt auch hier der Fokus weniger auf effizienter oder ausgefallener Programmierung.

```
# Append % sign

htmlTableHead = paste0(
"<table>",
  "<tr>",
    ,<th>Geschlecht</th>",
    "<th>Wahrscheinlichkeit</th>",
    "<th>Anzahl</th>",
  "</tr>",
"<tr>"
)
maleRow = paste0(
  "<tr>",
    "<td>M&aumlnnlich</td>",
    "<td>",shareMaleFinal,"</td>",
    "<td>",male,"</td>",
  "</tr>"
)
femaleRow = paste0(
  "<tr>",
    "<td>Weiblich</td>",
    "<td>",shareFemaleFinal,"</td>",
    "<td>",female,"</td>",
  "</tr>"
)
neuterRow = paste0(
  "<tr>",
    "<td>Neutrum</td>",
    "<td>",shareNeuterFinal,"</td>",
    "<td>",neuter,"</td>",
  "</tr>"
)
htmlTableBottom = "</table>"

htmlTableGender = paste0(
  htmlTableHead,
  maleRow,
  femaleRow,
  neuterRow,
  htmlTableBottom
)
```

Quellcode 5.8: Beispielcode in R zur Integration von Ergebnissen in eine HTML Tabelle

Die Variable $htmlTableGender$ kann nun in das Frontend eingesetzt werden. Da der HTML Code bereits funktionsfähig ist, kann er ebenfalls direkt implementiert werden. Im Backend,

d.h. in der $server.R$ Datei, wird zuerst Quellcode 5.7 in eine Funktion umgewandelt, welche die KWIC als Eingabeliste entgegennimmt und den HTML Code aus dem vorherigen Quellcode ausgibt. Danach wird diese Funktion in eine reaktive Funktion integriert. Diese erlaubt es, die übergebene Funktion jedes Mal auszuführen, sobald sich der Inhalt der Eingabemaske durch Benutzereingaben ändert. Die HTML Tabelle wird in die $output$ Liste übergeben, die dem Frontend zum Rendern der Ergebnisse zur Verfügung gestellt wird. Das Frontend wandelt den HTML Code in eine für Menschen lesbare Webseite um. Der folgende Quellcode 5.9 zeigt diesen Vorgang. Die Unterteilung in Front- und Backend wird über die Kommentare symbolisiert und um die Lesbarkeit zu erhöhen, wurden die für eine lauffähige Lösung notwendigen Programmteile ausgelassen und durch [...] ersetzt.

```
# server.R
shinyServer(function(input, output) {
[...]
findGenderOfNoun = reactive({genderFunction(input$kwic)})
output$gender = renderUI({findGenderOfNoun()})
[...]
})

# ui.R
[...]
htmlOutput("gender")
[...]
```

Quellcode 5.9: Integration der vorherigen Codes in ein R Shiny Frontend

Mit diesem Beispiel wird gezeigt, dass in wenigen Schritten lauffähige Funktionen in R Shiny programmiert werden können. Diese Funktionen müssen, um als Erweiterung von HanConc zu fungieren, nur noch an die entsprechenden Stellen in HanConcs Quellcode eingefügt werden.

Durch die offene und einfache Programmierung ist HanConc nicht nur ein linguistisches Werkzeug, dass ausschließlich von Programmierer_innen weiterentwickelt werden kann, sondern eine pädagogische Plattform, an der Linguist_innen, Schreibberater_innen und Studierende gleichermaßen arbeiten können. Ebenso kann HanConc an die Gegebenheiten und Ansprüche einzelner Einrichtungen angepasst werden und somit als pädagogisches Instrument eine korpusgetriebene Schreibberatung ermöglichen.

5.3.5 Textaufbereitung zu Term-Dokumenten Matrizen für Bag-of-Words basierte Funktionen in HanConc

Die im vorherigen Kapitel aufgezeigten Strukturen eignen sich gut zum Durchsuchen und Wiedergeben von Wörtern auf Zeichenebene. Das bedeutet, dass „Schule" und „Schüler" sich semantisch zwar ähnlich sind, jedoch in der eben dargestellten Struktur genauso verschieden sind wie „Schule" und „Dreipunktbiegeversuch". Um semantische Analysen in HanConc integrieren zu können, soll eine Latent Semantic Analysis (LSA) verwendet werden (siehe Kapitel 5.3.8.5).

Dieses Kapitel zeigt die notwendigen Aufbereitungsschritte für eine LSA. Eine statistische und linguistische Erklärung der einzelnen Schritte befindet sich in Kapitel 5.3.8.5.

Grundlage einer LSA ist eine Term-Dokumenten Matrix (TDM)[52]. Hierbei handelt es sich um ein Bag-of-Words Verfahren, bei dem syntaktische Zusammenhänge ignoriert werden. Die Integration der Texte wird mittels des lsa R Pakets durchgeführt (Wild 2015).

Als Basis für eine TDM werden die einzelnen Texte eines Korpus' als jeweils eigene Datei in einem Ordner gespeichert. Gegebenenfalls müssen die Texte noch in das TXT Format übertragen werden und Inhalte, die nicht UTF-8 kodiert sind, entfernt werden. Über die *textmatrix* Funktion werden die Texte je Korpus eingelesen, aufbereitet und sprachspezifische Stopwords[53] und Satzzeichen entfernt. Ebenso werden alle Wörter in Kleinbuchstaben umgewandelt. Ohne diese Umwandlung würden etwa Adjektive, wenn sie am Satzanfang stehen, als unterschiedliche Types behandelt. Es wird auf ein Stemming[54] verzichtet, um sicherzugehen, dass die Suchbegriffe aus einer Suche mit HanConc auf die richtigen Begriffe in der TDM gemapt werden. Im letzten Aufbereitungsschritt werden die erstellten Matrizen entsprechend der Länge der zugrundeliegenden Texte gewichtet. Hierzu wird das Term Frequency-Inverse Document Frequency (TF-IDF) Verfahren verwendet (Francis & Flynn 2010). Die Suche nach semantisch ähnlichen Wörtern erfolgt über den Vergleich der Wort-Vektoren, wobei die Vektoren semantisch ähnlicher Wörter hoch miteinander korrelieren. Firths Definition einer Kollokation „You shall know a word by the company it keeps!" bleibt damit gewahrt (Evert 2005).

Die so erstellten Matrizen werden bei Erstellung der Korpora vorberechnet und als R Data Dateien auf die Festplatte geschrieben. Bei Initialisierung von HanConc werden sie dann in den RAM geladen und zur Laufzeit von HanConc verwendet.

Die LSA soll verwendet werden, um den Benutzer_innen weitere semantisch ähnliche Suchbegriffe vorzuschlagen. Übliche Vorschlagssysteme beruhen auf dem historischen Verhalten anderer Nutzer_innen (Cambria, Schuller, Xia & Havasi 2013, Witten, Frank & Hall 2011). Onlineshops etwa nutzen entsprechende Algorithmen, um Kund_innen, die zum Beispiel eine Bohrmaschine kaufen wollen, auch noch die entsprechenden Bohrköpfe zu empfehlen. Da im Falle von HanConc die Anzahl an möglichen Wortkombinationen exponentiell zur Anzahl der Wörter im Korpus wächst und es im Vergleich dazu kaum Nutzer_innen gibt, wird auf einen solchen Ansatz verzichtet und stattdessen der bestehende Datensatz mit Hilfe einer LSA ausgewertet.

In der aktuellen Version von HanConc werden die einzelnen TDM nur für die Vorschlagssysteme auf LSA Basis genutzt, wobei weitere Nutzungsmöglichkeiten durchaus denkbar sind (Manning & Schutze 1999, Hofmann 2001, Stamatatos 2009, Francis & Flynn 2010).

[52] Im Zusammenhang mit einer LSA wird von Term-Dokumenten Matrizen und Dokument-Term Matrizen gesprochen. Um eine Matrix in die entsprechend andere zu überführen, müssen nur die Zeilen und Spalten transponiert werden.

[53] Stopwords sind Hochfrequenztermini, die grammatikalisch notwendig sind, jedoch wenig semantischen Beitrag leisten. Im Deutschen fallen etwa Artikel, Präpositionen und Konjunktionen in diese Kategorie.

[54] Beim Stemming werden morphologisch-grammatikalische Bestandteile von Wörtern, etwa Pluralendungen, entfernt.

5.3.6 Einsatz von HanConc auf verschiedenen Systemen

HanConc kann auf drei Arten bedient werden[55]. Bei R Shiny, auf dem das Interface von HanConc basiert, handelt es sich um ein HTML/JavaScript Webinterface für die Programmiersprache R. Dementsprechend kann HanConc auf einem RStudio- und Webserver ausgeführt und ebenfalls über einen modernen Browser aufgerufen werden. Alternativ kann R Shiny auch lokal in einem RStudiobrowser oder auf dem Lokalhost über einen modernen Browser geöffnet werden. Die HanConc Funktionen können auch ohne Graphical User Interface (GUI) bedient werden. Das Interface ist in diesem Fall der R Interpreter.

Erfolgt die Bedienung über das RStudio- oder Webinterface, ruft dieses über eine Schnittstelle die Shiny App auf. Diese App besteht aus einer Datei, die das Userinterface ($ui.R$) beschreibt, und einer Datei, welche die Funktionsaufrufe ($server.R$) steuert. Die $server.R$ Datei wiederum ruft die datenverarbeitenden Funktionen auf. Diese greifen auf die in Kapitel 5.3.3 beschriebenen Daten zurück. Die $ui.R$ Datei funktioniert ähnlich. Sie ruft ihrerseits weitere Elemente wie Funktionen, JavaScript, HTML und CSS auf. Die Shiny App verteilt die Funktionsaufrufe des Webinterfaces auf die $ui.R$ und die $server.R$ Datei. Die $ui.R$ Datei übergibt HTML und CSS Elemente, während die $server.R$ Datei R Funktionen aufruft. Diese Funktionen können Daten laden, diese verarbeiten und das Ergebnis an die $server.R$ Datei weiterleiten. Diese übergibt die Ergebnisse an die $ui.R$ Datei, welche wiederum die Ergebnisse in HTML Elemente umwandelt. Die $server.R$ Datei kann sowohl R Objekte als auch HTML Elemente übergeben. Gegebenenfalls können sowohl von der $ui.R$ Datei als auch von der $server.R$ Datei zusätzliche JavaScript Funktionen geladen und ausgeführt werden.

Shiny Apps können sowohl statisch als auch interaktiv gestaltet werden. Das heißt, dass sie ohne Interaktionsmöglichkeiten Ergebnisse von R Funktionen darstellen können oder aber auf Benutzer_inneneingaben reagieren können. Shiny bietet für letzteren Anwendungsfall eine Vielzahl an vordefinierten Eingabemethoden an[56].

HanConc ist als interaktive Shiny App gestaltet. Dies bedeutet, dass es im Frontend einige Elemente gibt, die durch Anwender_innen manipuliert werden können. Über Radio Buttons, Dropdown Listen und Texteingabefelder kann das gewünschte Korpus nach einer beliebigen Zeichenkette durchsucht werden. Um Fehleingaben zu minimieren, wurde HanConc mit einem Action Button versehen. Erst wenn der Action Button betätigt wird, schickt das Frontend die Anfrage an das Backend. Im Backend wird die Anfrage als R Liste mit dem Namen *input* übergeben. Die Elemente des Frontends werden unter ihrer inputId übergeben[57].

Auch wenn HanConc primär als Webapplikation gestaltet wurde, ist es ebenso möglich und für wissenschaftliche Zwecke auch erwünscht, direkt auf die Daten und den Quellcode zuzu-

[55] HanConc kann als vierte Alternative auch in einem Docker Container ausgeführt werden. Diese Alternative wird an dieser Stelle jedoch nicht diskutiert, da sie für die Intention, HanConc als pädagogisches Werkzeug in Schreibberatungen vorzustellen, nur aus technischer Sicht relevant ist. Für interessierte Leser_innen kann jedoch auf folgender Webseite eine Anleitung gefunden werden: https://www.r-bloggers.com/deploying-an-r-shiny-app-with-docker/ (Stand: 10. März 2020)

[56] https://shiny.rstudio.com/reference/shiny/1.1.0/ (Stand: 10. März 2020)

[57] Es ist zu bedenken, dass es sich hierbei nicht um die HTML ID handelt, sondern um einen String, der bei der Programmierung der $ui.R$ Datei definiert wird. Eine HTML ID kann unabhängig davon vergeben werden und entsprechend mittels JavaScript und CSS manipuliert werden.

greifen. Um Missbrauch zu vermeiden, wird empfohlen, Dritten die Daten nur unter Aufsicht zur Verfügung zu stellen und lokale Computer und Server entsprechend abzusichern. Ein Zugriff über eine Schnittstelle oder Programmschnittstelle ist nicht vorgesehen.

Die folgenden Kapitel stellen Einsatzmöglichkeiten von HanConc auf einem lokalen Computer (siehe Kapitel 5.3.6.1), über die Kommandozeile von R (siehe Kapitel 5.3.6.2) und auf einem Server dar (siehe Kapitel 5.3.6.3). Der Einsatz von Cloudlösungen wird ebenfalls in Kapitel 5.3.6.3 thematisiert.

5.3.6.1 Aufruf von HanConc in RStudio oder im Browser auf einem lokalen PC

HanConc auf einem lokalen Computer auszuführen, ist die wohl einfachste Möglichkeit das Programm zu nutzen. Die einzige Voraussetzung dafür ist eine installierte Version von R. Für den ersten Aufruf ist eine Internetverbindung notwendig, um eventuell nicht vorhandene R Pakete nachzuladen. Nach dem ersten Ausführen von HanConc wird keine Internetverbindung mehr benötigt. HanConc läuft auf jedem Betriebssystem, das R unterstützt (gängigen Windows Versionen, gängigen Linux Distributionen und unter Vorbehalt auch Mac).

Es ist zu beachten, dass HanConc als R Programm alle Daten in den RAM lädt. Daher ist ein RAM entsprechend der Korpusgröße notwendig. Die als R Data Datei gespeicherten Korpora nehmen in etwa die doppelte Größe im RAM im Vergleich zur Festplatte ein.

HanConc kann direkt aus R oder einem Integrated Development Environment (IDE) gestartet werden. Hierzu muss nur die $initialise.R$ Datei in R geladen und die darin enthaltene $initialise$ Funktion aufgerufen werden. Der Pfad zum HanConc Verzeichnis ist das einzige notwendige Argument der Funktion. Alternativ kann HanConc auch über die mitgelieferten Skripte gestartet werden. Für Windows liegt die $start.bat$ und für Linux die $start.sh$ bei. Beide Skripte sind auf Windows 7 und Fedora 23 getestet worden[58].

Wird HanConc über RStudio gestartet, öffnet sich aus RStudio heraus ein Browserfenster. Alternativ kann auch jeder auf dem System installierte Browser genutzt werden. HanConc wird in diesem Fall über $localhost : 5963$ in der Adresszeile aufgerufen[59].

5.3.6.2 Aufruf von HanConc über die Kommandozeile

Dieses Kapitel beschreibt wie einzelne Elemente von HanConc über die Kommandozeile genutzt werden können. Soll lediglich statt einer IDE die Kommandozeile genutzt werden, um HanConc zu starten, so muss nur die $initialise.R$ Datei geladen und mit dem entsprechenden Pfad ausgeführt werden. Dieses Vorgehen entspricht dem im vorherigen Unterkapitel beschriebenen Vorgehen.

HanConc ist zwar als Shiny App entwickelt worden, jedoch modular aufgebaut. Dementsprechend können und sollen auch alle Daten und Funktionen außerhalb von HanConc ver-

[58] Beide Skripte lesen das Verzeichnis aus, in dem sie sich befinden, starten R und übergeben diesen Pfad an die $initialise.R$ Datei. Sofern R einen Verweis in der $PATH$ Variable hat, spricht nichts dagegen, dass die Skripte auch auf nicht getesteten Systemen funktionieren. Mac Betriebssysteme sind nicht getestet worden.

[59] Wurde die $initialise.R$ Datei manipuliert, um HanConc an die lokalen Gegebenheiten anzupassen, kann sich der Port entsprechend ändern.

wendet werden. Quellcode 5.5 (Seite 119) zeigt beispielhaft, wie HanConc auch für andere Forschungszwecke verwendet werden kann.

Abbildung 5.13 (Seite 108) zeigt die backendseitigen Funktionsaufrufe von HanConc. All diese Funktionen werden vom $server.R$ Skript aufgerufen, sobald HanConc gestartet wird. Durch die weitgehende Unabhängigkeit der einzelnen Funktionen wird gewährleistet, dass sie für andere Projekte wiederverwendet werden können, ohne Abhängigkeiten berücksichtigen zu müssen.

Auch wenn die $server.R$ Datei mehr als 500 Zeilen Quellcode enthält[60], ist sie dennoch recht simpel aufgebaut. Die Korpora bestehen aus Listen mit DataFrames. Jedes DataFrame repräsentiert einen Satz und besteht wiederum aus drei Spalten. Die erste Spalte enthält alle Wörter je Satz. Die zweite Spalte besteht aus den Grundformen des jeweiligen Wortes, bzw. bei Sprachen, für die es keinen Lemmatiser gibt, aus den durch einen Stemmer gekürzten Wörtern oder den ursprünglichen Wörtern mit Kleinbuchstaben. Die dritte Spalte enthält die PoS Tags je Wort. Die Shiny Server Funktion ruft die KWIC Funktion auf. Diese nimmt die Suchparameter des Shiny Frontends auf und durchsucht das jeweilige Korpus. Die Ergebnisse werden in eine Liste aus DataFrames überführt und an die $results$ Funktion übergeben. Die $results$ Funktion berechnet alle relevanten Metriken und generiert die HTML Antwort für das Frontend. Wird HanConc als Ganzes verwendet, so wird das Ergebnis nun im Browser angezeigt. Soll HanConc allerdings für Forschungszwecke in anderen Funktionen wiederverwendet werden, so sollte an dieser Stelle die $results$ Funktion so manipuliert werden, dass sie nicht alle Ergebnisse als HTML zurückgibt, sondern die relevanten Teilergebnisse als R Objekte. Ebenso ist es in die andere Richtung möglich, die $results$ Funktion und das zurückgegebene HTML durch eigenen Code zu erweitern.

Kapitel 5.3.4 hat an zwei Beispielen gezeigt, wie eigene Funktionen auf Basis von HanConc programmiert werden können. Die zweite Beispielfunktion aus ebenjenem Kapitel, die das grammatikalische Geschlecht des Suchbegriffs ermittelt, kann direkt in HanConc integriert werden. Quellcode 5.6 (Seite 121) muss nur zu einer Funktion umgestaltet werden, die ein Substantiv und einen Satz als Input nimmt und die Gender Counts als Output zurückgibt. Diese Funktion kann nun in die $results$ Funktion integriert werden. Der vorgeschlagene Output aus Quellcode 5.8 (Seite 125) kann ebenso nahtlos zum HTML Code hinzugefügt werden.

5.3.6.3 Aufruf von HanConc im Browser auf einem Webserver

Das gesamte Backend von HanConc mit allen Funktionen ist in R geschrieben (R Core Team 2021). Das Frontend besteht aus R, HTML, JavaScript und CSS Code. Um den Code in eine von einem anderen Endgerät konsumierbare Weboberfläche zu verwandeln, wird ein Webserver benötigt. An dieser Stelle kommt, wie zuvor bereits beschrieben, Shiny der Firma RStudio[61] zum Einsatz. Shiny beantwortet die Anfrage eines Webbrowsers an den Webserver mit einer HTML Seite und dort eingebettetem JavaScript und CSS. Werden bei dem HTTP GET Request

[60] Werden nur die Zeilen betrachtet, die nicht aufgrund der in Kapitel 5.3.7 beschriebenen Einstellungsmöglichkeiten im Frontend benötigt werden, so reduziert sich die Zeilenanzahl massiv.

[61] https://shiny.rstudio.com/ (Stand: 10. März 2020)

weitere Benutzer_inneneingaben mitgegeben, so übergibt Shiny diese Parameter an die Funktionen im R Backend und sendet die Ergebnisse wieder zurück. RStudio stellt hierbei mehrere Möglichkeiten zur Verfügung, wie Shiny auf einem Webserver ausgeführt werden kann.

RStudio bietet mit ShinyApps.io einen Webservice für Shiny Apps an. Dieser Webservice erlaubt es, Shiny Apps ohne eigene Hardware zu betreiben. Hierbei wird lediglich der R Code hochgeladen und die Shiny App konfiguriert. Den eigentlichen Betrieb übernimmt RStudio. Die zugrunde liegende Infrastruktur wird von Amazon Web Services (AWS) in der Region „us-east-1" bereitgestellt[62]. Es ist daher zu bedenken, dass amerikanisches und nicht europäisches Recht gilt, sodass an dieser Stelle davon abgeraten werden muss, HanConc mit eventuell geschützten Texten ohne weitere rechtliche Prüfung auf ShinyApps.io zu hosten.

Als weitere Alternative zum Hosten einer Shiny Applikation richtet sich RStudio Connect an kommerzielle Nutzer_innen von Shiny. Im Gegensatz zu ShinyApps.io ermöglicht RStudio Connect das Hosten von Shiny Apps auf eigener Hardware. Neben dem eigentlichen Deployment von Shiny Apps stehen Funktionen zur Verfügung, die auf den Einsatz in Unternehmen und weniger auf private Endkonsument_innen oder einzelne Forscher_innen ausgerichtet sind. Das Hauptaugenmerk von Shiny Connect liegt daher auf den Anforderungen an Betriebsstabilität, IT Sicherheit und dem Arbeiten an Shiny Apps mit mehreren Entwickler_innen[63]. Aus Gründen der Preisgestaltung von RStudio Connect wird diese Lösung nicht weiter diskutiert[64].

Neben den beiden kostenpflichtigen Möglichkeiten Shiny Apps zu hosten, bietet RStudio auch eine kostenlose Variante an. Der Shiny Server kann entweder als kompilierte Installationsdatei heruntergeladen[65] oder aus den Quelldateien selbst kompiliert werden[66]. Es ist dabei zu bedenken, dass der Shiny Server nur für Linuxserver zur Verfügung steht. Da der Quellcode offen vorliegt, können selbst Anpassungen vorgenommen werden. Vor allem Sicherheitsfunktionen und Möglichkeiten zur Skalierung der Hardware sind in der kostenlosen Variante jedoch beschränkt. Abhilfe kann hier die Integration von Shiny in einen anderen Webserver wie Django für Python, Apache Tomcat für Java[67] oder ein Deployment in Docker Container[68] schaffen. Vor allem die Kombination aus Docker und einem Webserver kann dabei helfen, diese Limitierungen zu umgehen. Wird der bevorzugte Webserver als Reverse Proxy eingesetzt und Shiny aus Docker Containern heraus ausgeführt, so können die Sicherheitsfunktionen des Webservers mit der Skalierung der Docker Container kombiniert werden. Auf eine tiefergehende technische Diskussion wird mit Hinblick auf den Fokus dieser Arbeit verzichtet.

[62] https://docs.rstudio.com/shinyapps.io/security-and-compliance.html (Stand: 10. März 2020)

[63] https://docs.rstudio.com/connect/admin/ (Stand: 10. März 2020)

[64] Entsprechend der Preisliste betragen allein die Lizenzkosten mindestens $15.000 zuzüglich der Hardware und den damit verbundenen Kosten (https://rstudio.com/pricing/ (Stand: 20. März 2020)).

[65] https://rstudio.com/products/shiny/download-server/ (Stand: 10. März 2020)

[66] https://github.com/rstudio/shiny-server.git (Stand: 10. März 2020)

[67] https://support.rstudio.com/hc/en-us/articles/213733868-Running-Shiny-Server-with-a-Proxy (Stand: 10. März 2020)

[68] https://hub.docker.com/r/rocker/shiny (Stand: 10. März 2020)

5.3.7 Möglicher Input in das Frontend von HanConc

Nutzer_innen von HanConc können das Frontend in drei Dimensionen manipulieren: Es können sowohl ein oder mehrere Korpora ausgewählt werden, als auch die Suchparameter und Ergebnistypen beliebig komplex gestaltet werden. Die folgenden Unterkapitel stellen jeweils die Möglichkeiten in Bezug auf das Frontend vor und verweisen auf die Kapitel, die Hinweise zu den einzelnen Kopora geben, die Funktionen im Backend beschreiben oder die statistischen Hintergründe erläutern.

5.3.7.1 Auswahl der zu durchsuchenden Korpora

Das zu durchsuchende Korpus wird in HanConcs Frontend über ein Dropdown-Menü ausgewählt. Zum Zeitpunkt der Veröffentlichung dieser Arbeit befinden sich folgende Korpora in HanConc:

- Bauingenieurwesen / FBau
- Elektrotechnik und Informatik / FElt
- Maschinenbau / FMas
- Mathematik / FMat
- Naturwissenschaften[69] / FNat
- Philosophie[70] / FPhi
- Wirtschaftswissenschaften / FWir

Andere oder zusätzliche Korpora können jederzeit unter Beachtung der Hinweise in Kapitel 4 in HanConc integriert werden. Hierzu müssen lediglich die $ui.R$, $setcorpus.R$ und $setup.R$ Skripte angepasst werden. Die notwendigen Codeabschnitte befinden sich jeweils am Anfang der Skripte.

Wird die Checkbox „Enable Second Corpus" nicht ausgewählt, so beziehen sich alle Suchparameter und Ergebnistypen auf das ausgewählte Korpus. Sobald die Checkbox ausgewählt wird, kann über ein zweites Dropdown Menü ein zweites Korpus ausgewählt werden. Die Suchparameter beziehen sich dann auf beide Korpora. Über die Ergebnistypen „Graphs" und „Statistics" kann ein quantitativer Vergleich vorgenommen werden. Es wird bewusst auf den Einsatz von weiteren statistischen Vergleichen verzichtet, denn im Falle von nicht-parametrischen Verfahren wie χ^2 oder Kolmogorov-Smirnov Tests kann nicht automatisiert überprüft werden, ob es sich um statistische Artefakte handelt (Sheskin 2003). Gleiches gilt für parametrische Verfahren wie T-Tests. Bei diesen kommt erschwerend hinzu, dass die jeweiligen Voraussetzungen, wie eine zugrunde liegende Normalverteilung, ebenfalls erfüllt sein müssen.

[69] Das Korpus ist in HanConc enthalten, jedoch nicht im Frontend freigeschaltet, da das Korpus sehr groß ist und dementsprechend nur auf Computern oder Servern mit mehr RAM geladen werden kann.

[70] Das Korpus ist in HanConc enthalten, jedoch nicht im Frontend freigeschaltet, da das Korpus sehr groß ist und dementsprechend nur auf Computern oder Servern mit mehr RAM geladen werden kann.

Es ist nicht vorgesehen, für unterschiedliche Korpora verschiedene Suchparameter oder Ergebnistypen gleichzeitig auszuwerten. Solche Anfragen können durch den Einsatz mehrerer Browser-Tabs oder das Abspeichern von Zwischenergebnissen abgebildet werden. Allerdings ist es möglich, über die erweiterte Suche[71] zwei Suchbegriffe miteinander zu vergleichen. Hierbei wird jeweils eine Auswertung pro Suchbegriff erzeugt und diese Ergebnisse einander gegenübergestellt, sodass etwa Kollokationen miteinander verglichen werden können.

5.3.7.2 Suchparameter

Die Komplexität von Suchanfragen kann, wie eingangs erwähnt, in drei Dimensionen variiert werden. Suchanfragen, die über Einwortsuchen hinausgehen, müssen im Frontend über die Checkbox „Advanced Query" freigeschaltet werden. Zu jeder Parameterdimension wird im Folgenden vermerkt, wie diese im Frontend manipuliert werden kann.

In HanConc kann die Anzahl an gesuchten Wörtern auf bis zu fünf erhöht werden. Hierbei ist der Suchraum auf jeweils einen Satz begrenzt. Das heißt, Suchanfragen wie „Suche nach ‚Experiment' gefolgt von ‚Auswertung' im nächsten Absatz" sind nicht vorgesehen. Solch eine Suchanfrage muss selbst programmiert werden. Eine Anleitung hierzu befindet sich in Kapitel 5.3.4. Grundsätzlich werden solche Sätze zurückgegeben, in denen alle Suchbegriffe in der genauen Schreibweise vorkommen. Dies führt dazu, dass die Suchen nach „Experiment" und „Experimente" zu unterschiedlichen Ergebnissen führen. Auf Algorithmen, die ungenaue Suchen zulassen, wurde an dieser Stelle verzichtet. Die Levenshtein Distanz würde zum Beispiel eine Suche erlauben, in der „Experiment" und „Experimente" zum gleichen Ergebnis führen (Mahlow, Grün, Holupirek & Scholl 2012). Gleiches kann über eine Suche mittels RegEx erreicht werden[72]. Allerdings wurde zur Reduzierung der Komplexität des Codes und der Laufzeit von Suchen in HanConc hierauf verzichtet. Alle Ergebnisse wie N-Grams oder Kollokationen, die über KWIC hinausgehen, beziehen sich auf den ersten Suchbegriff. Wenn also der erste Begriff „Experiment" und der zweite „durchführen" ist, so werden zwar alle Sätze mit diesen Suchbegriffen zurückgegeben, jedoch nur die Kollokationen für „Experiment".

Die Suche kann um eine zweite Dimension erweitert werden. Hierbei werden zusätzlich zu den tatsächlichen Wörtern noch weitere Annotationen zur Suche hinzugezogen, sodass nach den gestemmten oder lemmatisierten Wörtern, inhaltlichen Annotationen oder Part-of-Speech (PoS) Tags gesucht werden kann. Wie in Kapitel 5.3.3 beschrieben, ist für die Umsetzung in HanConc eine mehrfach geschachtelte Datenstruktur notwendig: jedes einzelne Wort wird in einer Zeile in einem DataFrame je Satz in einer Liste je Text und wiederum in einer Liste pro Korpus dargestellt. Die erste Spalte des DataFrames enthält die tatsächlichen Wörter, wie sie im ursprünglichen Text vorkommen. Die zweite Spalte beinhaltet die gestemmten oder lemmatisierten Wörter der ersten Spalte und ist explizit für alternative Beschreibungen der Ursprungswörter vorgesehen. Vor allem für Sprachen außerhalb des Englischen sind selten qualitativ hochwertige Lemmatisierer vorhanden (Pütz, De Kok, Pütz & Hinrichs 2018, Müller, Cotterell, Fraser

[71] Die erweiterte Suche kann über die Checkbox „Advanced Query" aktiviert werden.
[72] Hierzu müsste nach „Experiment[a-z]{*}" gesucht werden.

& Schütze 2015). In diesen Fällen kann beispielsweise eine Named Entity Recognition (NER) eingesetzt werden, um etwa Fachbegriffe, Personen-, Firmen- und Ortsnamen zu identifizieren und suchbar zu machen (Chiu & Nichols 2016, Derczynski, Maynard, Rizzo, Van Erp, Gorrell, Troncy, Petrak & Bontcheva 2015). In der dritten Spalte befinden sich die PoS Tags.

Die bisher beschriebenen Suchen beschränkten sich auf die erste Spalte des DataFrames. Über die Checkbox „Advanced Query" kann die Suche auf Lemmata und PoS Tags erweitert werden. Wie oben ebenfalls erwähnt, können bis zu fünf Elemente gesucht werden. Bei diesen fünf Elementen können Wörter, Lemmata und PoS Tags kombiniert werden, wobei mindestens ein Wort aus der ersten Spalte enthalten sein muss. Diese Limitierung ist als Sicherheitsmaßnahme eingebaut worden, um zu verhindern, dass zu breite Suchen, wie nach Sätzen, die einen Artikel und eine Präposition enthalten, den PC bzw. Server überlasten. Auch hier gilt wieder, dass sich komplexere Ergebnistypen nur auf das erste Wort beziehen.

Um nach komplexeren Satzkonstruktionen zu suchen, kann die Option „Order matters" ausgewählt werden. Diese Checkbox sorgt dafür, das bei einer Suche mit mehreren Elementen die Reihenfolge als verbindlicher Parameter berücksichtigt wird. Ob es sich bei den Suchelementen um Wörter, Lemmata oder PoS Tags handelt, ist in diesem Fall irrelevant. Somit kann etwa folgende Suche abgebildet werden:

'das' + 'Experiment' + 'wurde' + VVPP

Mit dieser Suche kann ein passendes Partizip Perfekt im Zusammenhang mit dem Wort „Experiment" gefunden werden. Jedes Suchelement wird durch eine eigene Farbe in den KWIC markiert. Somit wird etwa bei dem obigen Beispiel auf den ersten Blick ersichtlich, welches tatsächliche Wort sich hinter dem PoS Tag „VVPP" verbirgt.

An dieser Stelle muss darauf hingewiesen werden, dass diese Art der Suche zu hohen Laufzeiten führt. Nachdem alle Sätze mit „das" zu einer Liste hinzugefügt wurden, wird diese Liste anchließend auf solche Sätze reduziert, welche die weiteren Kriterien erfüllen. Die primäre Ergebnisliste ist wesentlich größer, als jene, die nur Sätze mit dem Wort „Experiment" enthält. Wird das seltenere Wort als primäres Suchkriterium gewählt, kann die Laufzeit deutlich reduziert werden. Die folgende Suche hat daher eine deutlich kürzere Laufzeit als das obige Beispiel:

'Experiment' + 'wurde' + VVPP.

5.3.8 Möglicher Output von HanConc

Als pädagogisches Werkzeug für Schreibberater_innen und ihre Studierenden soll HanConc mehr Funktionen anbieten können, als nur Beispielsätze zu Suchanfragen anzuzeigen. Deshalb bietet HanConc Ergebnistypen, die über KWIC deutlich hinausgehen. In diesem Kapitel werden alle Ergebnistypen beschrieben, die bei einer Suche mit HanConc zurückgegeben werden können. Dabei wird jeweils auf die linguistischen und statistischen Hintergründe eingegangen. Sollten die Ergebnisse nicht direkt aus den Korpora erzeugt werden können, d.h. andere Vorbereitungsschritte oder Datengrundlagen notwendig sein, wird explizit darauf hingewiesen.

Jeder Ergebnistyp wird folgendermaßen beschrieben: Zunächst wird der notwendige Input definiert, woraufhin eine linguistische Definition beschrieben wird, um anschließend gege-

benenfalls auf die statistischen Grundlagen bzw. die jeweiligen Machine Learning Verfahren einzugehen. Im Weiteren wird eine exemplarische Darstellung im Frontend vorgestellt und abschließend auf die mögliche Nutzung in einer Schreibberatung hingewiesen.

Da die didaktische Umsetzung von Korpuslinguistik in Schreibberatungen nicht Fokus dieser Arbeit ist, werden tatsächliche Einsatzmöglichkeiten der Ergebnistypen lediglich skizziert. Eine tiefergehende Analyse und Beschreibung von didaktischen Konzepten soll entsprechenden Arbeiten überlassen werden.

5.3.8.1 Key Words in Context

0. Einleitung

Bei den KWIC handelt es sich um den primären und originären Ergebnistyp aller bisher beschriebenen Konkordanzprogramme und daher auch von HanConc. KWIC bestehen aus dem Suchwort und dem umgebenden Kontext. Wie weit Kontext definiert ist, hängt von der verwendeten Software bzw. den Buchautor_innen der linguistischen Fachliteratur ab. In HanConc wird eine syntaktische Definition von Kontext verwendet, was bedeutet, dass das Ende eines Satzes den Kontext[73] limitiert. Alternativen wie WordSmith Tools (WST) und AntConc hingegen definieren den Kontext anhand einer Anzahl an Zeichen zur Rechten und Linken des Suchworts.

Außerhalb der Linguistik sind Konkordanzen und KWIC teils seit Jahrhunderten im Einsatz. Die erste Bibelkonkordanz, die zu Hauptbegriffen der Bibel die entsprechende Bibelstelle und deren Kontext angibt, wird auf das Jahr 1252 datiert (Calwer Verlag 1979). In der Pädagogik werden Korpora seit den 1980ern verwendet. Die ersten pädagogischen Korpusanalysen wurden dabei noch mit spezifisch zusammengestellten und händisch ausgewerteten Korpora in kleineren Kontexten durchgeführt (Römer 2006). Erst datengetriebenes Lernen (Chitez, Rapp & Kruse 2015), eine große Anzahl an kostenlos oder kostengünstig zur Verfügung stehenden Texten (Römer 2006), entsprechende kostengünstige Rechenleistung und kostenlose Software wie AntConc (Anthony 2019) haben zur weiteren Verbreitung von Korpora und KWIC in pädagogischen und korpuslinguistischen Fachkreisen und darüber hinaus geführt.

1. Input

Als Input für die KWIC dienen die in Kapitel 5.3.3.3 vorgestellten R Datenobjekte. Diese werden entsprechend der aus dem Frontend übergebenen Parameter durchsucht und die Ergebnisse in ein neues Objekt geschrieben. Wird nur nach einem Wort gesucht, so gibt die KWIC Funktion die erste Spalte jedes Dataframes zurück, die das Suchwort enthält. Wird nach mehreren Wörtern, Lemmata oder PoS Tags gesucht, werden auch die übrigen Spalten berücksichtigt. Wird beispielsweise nach einem Artikel gesucht, so iteriert der Algorithmus zusätzlich über die dritte Spalte jedes Dataframes, sucht nach der Zeichenkette „ART" und übergibt das Dataframe an die Ergebnisliste, falls ein Artikel gefunden wurde (siehe auch Quellcode 5.10).

```
1 for(text in corpus){
2     for(sentence in text){
3         for(word in sentence){
```

[73] Der Ergebnistyp „Larger Context" weicht diese Limitierung auf und gibt auch einige umgebende Sätze zurück.

```
            if(corpus[text[sentence[word]]][,words] == searchword){
                results = append(results, corpus[text[sentence]])
            }
            if(searchpostag != null){
                if(corpus[text[sentence[word]]][,pos] == searchpostag){
                    results = append(results, corpus[text[sentence]])
                }
            }
        }
    }
}
```

Quellcode 5.10: HanConcs Suchalgorithmus als gekürzter R Pseudocode

Damit die Ergebnisse besser verständlich sind, werden sie unterschiedlich eingefärbt, sodass das erste Suchwort eine andere Farbe erhält als, wie in diesem Beispiel, die zusätzlich gefundenen Artikel. Die Einfärbung erfolgt mittels HTML Tags, die später vom Frontend gerendert werden:

Tabelle 5.7: Beispielsätze für KWIC Suchergebnisse mit farblichen Markierungen

(1) Das Experiment ist wunderbar verlaufen.
(2) Mit dem Experiment konnte gezeigt werden, dass eine Lösung gefunden werden kann.
(3) Trotz aller Anstrengungen konnte das Experiment nicht erfolgreich beendet werden.

2. Linguistik

Linguistisch dienen KWIC in Anlehnung an Firths Prinzip von „You shall know a word by the company it keeps" (Evert 2005) als Möglichkeit einer sprachlichen Einordnung von Fachbegriffen, die nicht ausreichend präskriptiv kodifiziert oder den Schreibberater_innen und ihren Studierenden unbekannt sind. Die KWIC Funktion kann eingesetzt werden, um ein geeignetes Korpus zu durchsuchen und sich ein Bild davon zu machen, in welchen Zusammenhängen der jeweilige Begriff verwendet wird. Die Vorgehensweise ist dabei umgekehrt zu einem Thesaurus oder Wörterbuch. Es soll nicht der passende Begriff für einen sprachlichen Kontext gesucht werden, sondern ein sprachlicher Kontext für einen notwendigen Begriff gefunden werden.

3. Statistik

Da es sich bei den KWIC um einen qualitativen Ergebnistyp handelt, ist eine statistische Beschreibung unnötig.

4. Darstellung im Frontend

KWIC sind eine grundlegende Funktion von Korpussoftware und werden daher als erster Ergebnistyp in der Auswahlliste von HanConcs Frontend verwendet. Die KWIC werden in HanConc bei jeder Suche dargestellt, indem je ein Satz pro Zeile in den Suchergebnissen angezeigt wird. Um die Ergebnisse zu strukturieren, verdeutlichen eine dünne Linie und ein kleiner Absatz die Grenzen der einzelnen Sätze. Werden mehrere Suchbegriffe eingegeben, so werden diese einzeln farbig markiert. Zusätzliche Ergebnistypen werden oberhalb der KWIC dargestellt. Der Schieberegler „Sentence Limit" ermöglicht es, die Anzahl an angezeigten Sätzen zu variieren.

Wird die Checkbox „No Sentence Limit" ausgewählt, so werden alle Sätze angezeigt. Dies kann jedoch bei einer großen Anzahl an Sätzen die Laufzeit negativ beeinflussen.

5. Nutzen in der Schreibberatung

In der Schreibberatung können KWIC explorativ und ohne eine konkrete These überprüfen zu wollen, eingesetzt werden, um einen Suchbegriff anhand mehrerer Beispiele zu analysieren. KWIC können ebenso als erster Schritt genutzt werden, um zu überprüfen, ob ein Suchbegriff überhaupt im entsprechenden Korpus vorkommt. Ebenso können KWIC genutzt werden, um Beispielsätze für andere Ergebnistypen zu finden.

5.3.8.2 Frequenz

0. Einleitung

Die Grundlage jeder quantitativen Analyse ist das Auszählen von Beobachtungen[74]. Im Fall von HanConc wird die Anzahl an Sätzen ausgezählt, die den Suchkriterien entsprechen. Es handelt sich bei der Frequenz somit um eine absolute Häufigkeit. Sie kann genutzt werden, um einzuschätzen, ob es sich um ein seltenes und spezialisiertes Wort handelt oder um einen geläufigen Begriff in der jeweiligen Fachsprache. Die Frequenz bietet die konzeptionelle Grundlage für weitere Ergebnistypen. Erst eine grundsätzliche Akzeptanz eines quantitativen Ansatzes führt zur regelmäßigen Nutzung dieser Ergebnistypen.

1. Input

Die Frequenz basiert auf der Länge der Ergebnisliste der KWIC. Ein mehrfach im Satz verwendeter Begriff wird trotzdem nur einfach gezählt.

2. Linguistik

Da es sich bei der Frequenz um eine statistische Größe handelt, wird auf eine linguistische Beschreibung verzichtet.

3. Statistik

Die Frequenz ist als Grundlage für weitere statistische Analysen essentiell. Erst wenn die Anzahl an Beobachtungen groß genug ist, können darauf folgende Verfahren zu validen Ergebnissen führen. Entsprechend des Gesetzes der großen Zahlen kann es bei zu wenigen Beobachtungen zu verzerrten Ergebnissen kommen, da Ausreißer stärker ins Gewicht fallen (Wooldridge 2010). Baayen weist darauf hin, dass die Wortfrequenzen von der Korpusgröße abhängig sind und daher das Gesetz der großen Zahlen nicht uneingeschränkt zur Anwendung kommt (2001)[75]. Auch wenn Wortfrequenzen dem Zipf-Mandelbrot Gesetz folgen (Evert & Baroni 2007, Kirby 1985, Bentz, Kiela, Hill & Buttery 2014, Montemurro 2001) und daher weder gleich- noch normalverteilt auftreten, so muss doch, um Verzerrungen in fortgeschritteneren Verfahren zu

[74] Bortz et al. (2010) geben eine Einführung in quantitative Analysen und Forschung aus Sicht der Geisteswissenschaften.

[75] Googles NGram Viewer findet beispielsweise hunderte Vorkommen von der falschen Schreibweise „beleive" in englischsprachigen Büchern der letzten zwei Jahrhunderte, sodass der Eindruck entstehen könnte, dass „beleive" eine legitime Schreibweise sei. Allerdings ergeben sich die Vorkommen aus der enormen Korpusgröße und stellen somit ein statistisches Artefakt dar (https://books.google.com/ngrams/graph?content=beleive&year_start=1800&year_end=2019&corpus=en-2019&smoothing=0 (Stand: 15. Dezember 2022)).

verhindern, auf eine ausreichend große Anzahl an Beobachtungen bestanden werden.

4. Darstellung im Frontend

Die Frequenz wird in HanConc als Zahl unter der entsprechenden Überschrift in einer eigenen Zeile dargestellt.

5. Nutzen in der Schreibberatung

Neben den KWIC sollte die Frequenz der am häufigsten genutzte Ergebnistyp sein. Mit der Frequenz kann überprüft werden, ob es sich bei den KWIC um statistische Artefakte handelt oder die Wörter tatsächlich regelmäßig verwendet werden. Als Beispiel kann der Unterschied zwischen „believe" als korrekte Schreibweise und „beleive" als Tippfehler herangezogen werden. Im BNC gibt es neun Treffer für die falsche Schreibweise. Das heißt, wenn das Korpus nur groß genug ist, können auch falsche Schreibweisen zu Treffern führen. Wird jedoch auch die Frequenz als Kontrollinstrument herangezogen, so wird deutlich, dass es sich bei der zweiten Schreibweise um ein Artefakt handelt. Die korrekte Schreibweise führt zu 20.192 Treffern, was zeigt, dass die Frequenz einen Beitrag dazu leisten kann, tradierte von abweichenden Schreibweisen und Verwendungen zu unterscheiden.

Vor allem, wenn zwei Korpora oder zwei Wörter miteinander verglichen werden, sollte die Frequenz zum Einsatz kommen. Hierdurch kann bei semantisch ähnlichen Wörtern herausgefunden werden, welches die Leser_innen im Kontext eher erwarten würden. Ebenso kann für ein Wort untersucht werden, ob es innerhalb der verschiedenen akademischen Fachrichtungen zu unterschiedlichen Verwendungen des Wortes kommt.

Sollten komplexere statistische Verfahren zum Einsatz kommen, so ist es ebenfalls zwingend notwendig, die Frequenz zu überprüfen. Führt eine Suche jedoch nur zu wenigen Treffern, so ist von weiteren Verfahren abzusehen. HanConc hat keine fest hinterlegten Minimalanforderungen für weiterführende Ergebnistypen. Es liegt im Ermessen der Nutzer_innen, zu entscheiden, ob die Datengrundlage ausreichend ist, um zu verlässlichen und validen Ergebnissen zu kommen.

5.3.8.3 Kollokationen, N-Grams und der Mutual Information Score

0. Einleitung

Sprache ist mehr als die reine Aneinanderreihung von Wörtern. Die Beschreibung von Sprache und das Unterrichten von Fremdsprachen basieren auf Wortzusammenhängen und Kontext. Auch wenn, sofern grammatikalische Regeln eingehalten werden, grundsätzlich alle Wörter miteinander kombinierbar erscheinen, so gibt es doch Kombinationen, die als natürlicher als andere angesehen werden (Durrant & Schmitt 2009). Durrant weist darauf hin, dass es sprachspezifische Wortkombinationen gibt, die aus semantischen Gründen blockiert sind (Durrant & Schmitt 2009).

Um den akademischen Traditionen der Nutzer_innen Rechnung zu tragen, bietet HanConc für die Überprüfung der Natürlichkeit von Wortkombinationen sowohl Kollokationen als auch N-Grams als Ergebnistypen an. Kollokationen werden vor allem in der Linguistik verwendet (Kennedy 2003, Nesselhauf 2003, Hommerberg & Tottie 2007, Siyanova & Schmitt 2008,

Durrant 2010, Granger 2011, Paquot & Granger 2012), während N-Grams aufgrund der kürzeren Laufzeiten eher in der Informatik und für Big Data eingesetzt werden (Manning & Schutze 1999, Lin & Dyer 2010, Tsai 2011, Ellis 2012, Cambria et al. 2013)[76].

In diesem Kapitel werden Kollokationen und N-Grams trotz ihrer unterschiedlichen Traditionen gemeinsam behandelt, da sie in ihrer Programmierung große Ähnlichkeiten aufweisen und ähnlich verwendet und interpretiert werden können. Der MI Score wird als Interpretationshilfe für die Kollokationen ebenfalls in diesem Kapitel eingeführt.

1. Input

Sowohl die Kollokationen als auch die N-Grams können direkt aus HanConcs Suchergebnissen gewonnen werden. Der MI Score benötigt zusätzlich noch die Wortliste des jeweiligen Korpus'.

Die Kollokationen werden für die einzelnen Wörter, Lemmata und PoS Tags ausgewertet. Hierzu werden jeweils vier Vektoren erzeugt, wobei jeder Vektor die Kollokation an der zweiten und ersten Position links und rechts des Suchworts repräsentiert. Für jedes Suchwort werden dementsprechend folgende Positionen ausgewertet:

2. links	1. links	Suchwort	1. rechts	2. rechts
Wort	Wort	Suchwort	Wort	Wort
Lemma	Lemma	Suchwort	Lemma	Lemma
PoS Tag	PoS Tag	Suchwort	PoS Tag	PoS Tag

Position:	1	2	3	4	5	6	7	8
Wort:	An	dem	Bauteil	wurde	ein	Versuch	erfolgreich	durchgeführt.
PoS Tag:	APPR	ART	NN	VAFIM	ART	NN	ADJ	VVPP

Tabelle 5.8: KWIC Beispielsatz zur Verdeutlichung der Erzeugung von Kollokationen und N-Grams

Tabelle 5.8 zeigt für das Wort „Versuch" einen Beispielsatz inklusive der Positionsangaben und PoS Tags. Für die Kollokationen werden die Positionen zwei links bis zwei rechts vom Suchwort betrachtet. Für dieses Beispiel bedeutet dies, dass die Positionen vier, fünf, sieben und acht mit in die Kollokationen einfließen. HanConc fügt für die Kollokationen an der zweiten Positionen links vom Suchwort das Wort „wurde" und das PoS Tag „VAFIM" zu den entsprechenden Vektoren hinzu. Für die anderen Positionen wird ebenso verfahren. Ist ein Satz so gestaltet, dass eine der Positionen nicht besetzt ist, so wird auch kein Element an den Vektor angehängt.

Sobald alle Sätze ausgewertet wurden und die Elemente zu den Vektoren hinzugefügt wurden, werden diese ausgezählt. Das hierbei entstehende DataFrame enthält in der ersten Spalte das Element (Wort, Lemma oder PoS Tag) und in der zweiten Spalte die jeweilige Anzahl davon. Das Frontend von HanConc ist so ausgelegt, dass es die häufigsten fünf Elemente anzeigt. Bei dieser Darstellung wird das Element und die Anzahl in runden Klammern dargestellt. Es ist zu beachten, dass die Elemente weder semantisch noch syntaktisch zusammenpassen müssen.

[76]Dementsprechend basiert zum Beispiel auch Googles Suchmaschine für Wortpaare auf N-Grams (https://books.google.com/ngrams (Stand 26. August 2018)).

Das heißt, das häufigste Wort an der zweiten Position muss nicht zwangsläufig zu dem Wort an der ersten Position links passen. Gleiches gilt für die Lemmata und PoS Tags. Vor allem bei den PoS Tags kann es zu ungrammatikalischen Kombinationen kommen.

N-Grams werden ebenso mit den DataFrames der Ergebnisliste erzeugt. HanConc gibt Bi-Grams und Tri-Grams zurück[77]. Die Bi- und Tri-Grams werden durch Konkatenation von bis zu drei Elementen rund um den Suchbegriff erzeugt. Ebenso wie bei den Kollokationen werden sowohl Wörter, als auch Lemmata und PoS Tags berücksichtigt. Im Gegensatz zu Kollokationen werden jedoch nur tatsächliche Mehrwortkombinationen betrachtet. Bei den Tri-Grams wird von drei Fällen ausgegangen:

- Der Suchbegriff steht links.
- Der Suchbegriff ist in der Mitte.
- Der Suchbegriff steht rechts.

Je nach Fall werden bis zu zwei Elemente links, rechts oder links und rechts aufgefüllt. Im ersten Fall werden die beiden Elemente links des Suchbegriffs, im zweiten Fall je ein Element zu beiden Seiten und im dritten Fall die beiden Elemente rechts des Suchbegriffs betrachtet. Die Programmierung ähnelt hier der Programmierung für die Kollokationen. Je Fall werden drei Vektoren generiert, wobei je einer für Wörter, Lemmata und PoS Tags vorgesehen ist. Für jeden Fall werden wiederum die entsprechenden Elemente konkateniert und an den dafür vorgesehen Vektor angehängt. Die Vektoren werden letztlich ausgezählt und die häufigsten fünf N-Grams je Elementtyp dargestellt.

Der Mutual Information (MI) Score beschreibt die gegenseitige semantische Anziehungskraft der einzelnen Bestandteile einer Kollokation. HanConc benutzt den MI Score, um Kollokationen auf ihren idiomatischen Zusammenhalt zu überprüfen. Das folgende Statistikunterkapitel thematisiert die Vorteile des MI Scores gegenüber ausgezählten Kollokationen in Bezug auf ihre idiomatische Natürlichkeit. Die Programmierung des MI Scores basiert auf den Kollokationsdataframes und der Wortliste. Nachdem die Kollokationen wie oben beschrieben ausgewertet worden sind, wird zuerst die Häufigkeit des Suchbegriffs in der Wortliste und daraufhin jedes einzelne Wort, das mit dem Suchbegriff kollokiert, gesucht. Entsprechend der Formel, die im Statistikunterkapitel beschrieben ist, wird dann der MI Score für jede Kollokation berechnet und die Ergebnisse als zusätzliche Spalte in das DataFrame geschrieben. Im Frontend werden die fünf stärksten Kollokationen entsprechend dem MI Score dargestellt. Für N-Grams wird kein MI Score berechnet, da sie als ein Element gesehen werden und eine Berechnungsgrundlage nicht gegeben ist.

2. Linguistik

Kollokationen können auf drei Arten definiert werden. Halliday & Hasan (2014) definieren Kollokationen aus Diskursperspektive als Teil von Kohäsion. Kollokationen sind aus ihrer Sicht somit Wörter, die Kohäsion über mehrere Sätze herstellen können. Die einzelnen Elemente

[77] Bei Bi-Grams handelt es sich um Zweiwortkombinationen, während Tri-Grams drei Wörter enthalten.

haben dabei keinerlei Verbindung, außer dass sie im selben diskursiven Kontext vorkommen können. In ihrem Kapitel 6.4 stellen Halliday & Hasan dar, dass in bestimmten diskursiven Zusammenhängen etwa „child", „boy" und „girl" als Kollokation zu verstehen sind. Sie benutzen die Beispielsätze:

(1) Why does this little boy wriggle all the time? Girls don't wriggle.

In diesen Sätzen können „child", „boy" und „girl" beliebig ausgetauscht werden. Trotz unterschiedlicher Referenz können alle drei Elemente genutzt werden, um einen Zusammenhang zwischen den beiden Sätzen herzustellen, da sie zur gleichen lexikalischen Familie gehören. Sie fungieren daher als „cohesive force" (Halliday & Hasan 2014, 285). Die Autoren definieren die Abgrenzung noch weiter und kontextspezifischer, sodass „wallowing", „sinking", „buried" und „imbedded" in einem speziellen Diskursbeispiel von ihnen ebenfalls als Kollokation angesehen werden (Halliday & Hasan 2014, 287). Dieser Diskursansatz lässt sich allerdings kaum so operationalisieren, als dass sich hieraus eine Programmierungsstrategie ableiten ließe.

Eine zweite mögliche Definition von Kollokationen basiert auf semantischen Elementen in festgelegten grammatikalischen Konstruktionen. Diese Kollokationstypen bestehen aus zwei oder mehr Elementen, die nahe zueinander in einem Textabschnitt vorkommen. Leacock, Chodorow, Gamon & Tetreault (2014, 63) verwenden etwa das Beispiel einer „Verb plus Substantiv" Kollokation. Hierbei wird durch Konventionen festgelegt, welches Verb mit dem bestimmenden Substantiv verwendet werden kann. Deshalb wird Satz (1) als wohlgeformt akzeptiert, während Satz (2) abgelehnt wird (Leacock, Chodorow, Gamon & Tetreault 2014, 63):

(2) Hold an election.

(3) * Make an election.

Leacock et al. weisen darauf hin, dass solche Kollokationen durch Auszählen eines ausreichend großen Korpus' und entsprechende Signifikanztests ermittelt werden können. Durch diesen Ansatz kann außerdem auf eine Unterscheidung zwischen Idiom und Kollokation verzichtet werden, da ein Idiom auch nur eine entsprechend starke Kollokation ist.

Das obige Beispiel (2) gibt bereits einen Hinweis auf mögliche Probleme bei der Umsetzung in einer Korpussoftware im Allgemeinen und deutschsprachigen Fachwissenschaftstexten im Speziellen. Für die Programmierung einer Korpussoftware müsste unter anderem berücksichtigt werden, dass beliebig viele Elemente etwa zwischen dem Verb und dem Substantiv stehen können. Zusammengesetzte Verben oder eingeschobene Adjektive zum Beispiel verhindern, dass der Kollokationstyp „Verb plus Substantiv" über programmierte Regeln umgesetzt werden kann. Um also das passende Verb zu einem Substantiv innerhalb eines Satzes beziehungsweise innerhalb einer Nominalphrase zu finden, wäre ein syntaktischer Parser notwendig.

Ältere probabilistische Parser wie der Stanford Parser (Manning & Schutze 1999) (in den Versionen vor 2014) erlaubten zwar das automatisierte Erstellen von Syntaxbäumen, jedoch ist die Präzision besonders für nicht englische Texte unzureichend. Neuere Ansätze, welche eine deutlich höhere Präzision erreichen, basieren auf neuronalen Netzwerken wie den Recursive

Neural Networks der neueren Versionen des Stanford Parsers (Socher, Manning & Ng 2010). Als Alternative bietet sich Googles Parser an. Dieser basiert ebenfalls auf einem neuronalen Netzwerk. In diesem Fall handelt es sich um ein Long Short-Term Memory (LSTM) Netzwerk, das mit TensorFlow umgesetzt wurde (Andor et al. 2016). Ein deutsches Modell wurde zum Zeitpunkt, als diese Arbeit verfasst wurde, noch nicht trainiert.

Auf den Einsatz von syntaktischen Parsern und damit auf eine technische Umsetzung der Anforderungen, die sich aus der obigen Definition von Kollokationen ergeben, wurde aus mehreren Gründen verzichtet: Für die Verwendung des Stanford als auch des Google Parsers werden neben R entweder noch Java oder TensorFlow benötigt. Entsprechend der Idee, HanConc möglichst einfach und einsteigerfreundlich zu gestalten, werden keine Module genutzt, die andere Programmiersprachen als R benötigen, auf bestimmte Betriebssysteme festgelegt sind oder nur mit einigen Versionsnummern funktionieren. Außerdem benötigen geparsete Sätze eine andere Datenhaltung als Sätze, die von einem PoS Tagger annotiert wurden. Syntaktische Bäume können nicht als flache Tabellen gespeichert werden, sondern benötigen einen Datentyp, der dieser hierarchischen Struktur Rechnung trägt. R stellt mit seinen Listen einen solchen Datentyp zur Verfügung. Allerdings ist die Übertragung des Outputs des Stanford als auch des Google Parsers in das Zielformat und die Abfrage der R Liste ungleich komplexer als eine Suche mit den vorhandenen Satztabellen. Für eine solche Anwendung müsste ein neuer Suchalgorithmus programmiert werden, der nicht nur auf einer Ebene sucht, sondern auch die Ebenen darüber und darunter berücksichtigt. In Hinblick auf die Anforderungen an HanConc als einfach manipulierbare Korpussoftware wird die obige Definition von Kollokationen daher nicht umgesetzt.

Eine Kollokation kann auch als Kookkurrenz verstanden werden, bei welcher die zusammengesetzte beziehungsweise zusammen auftretende Wortform überproportional häufiger im Gegensatz zu den Einzelformen auftritt. Diese dritte Form der Kollokationsdefinition hat mehrere Vorteile: Sie ist, unabhängig vom semantischen Inhalt, leicht von einem Algorithmus zu finden. Ebenso können, im Gegensatz zum oben beschriebenen Diskursansatz, Kollokationen auf diese Weise eindeutig identifiziert werden, da die Entscheidung, ob es sich um eine Kollokation handelt, auf Basis von statistischen Auswertungen und nicht von Interpretationen getroffen wird. Da es sich um einen quantitativen und nicht um einen semantischen Ansatz handelt, müssen die Besonderheiten einzelner Fachdisziplinen nicht explizit berücksichtigt werden. Außerdem kann diese Definition von Kollokationen mit den bereits bestehenden Datentypen und Aufbereitungsschritten von HanConc umgesetzt werden (Biber 1993, Pearce 2001). Der Ansatz, Kollokationen als statistisch überhäufige Kookkurrenzen zu definieren, hat den entscheidenden Vorteil, dass mit N-Grams ein ähnliches Konstrukt aus Sicht der Informatik besteht, das bereits für diverse Übersetzungs- und Big Data Textanalysen verwendet wird (Wang, Utiyama, Goto, Sumita, Zhao & Lu 2016, Zamora-Martinez & Castro-Bleda 2018, Mohammad, Salameh & Kiritchenko 2016, Luong, Pham & Manning 2015, He, He, Wu & Wang 2016, Shao, Feng & Chen 2018, Ha, Cho, Niehues, Mediani, Sperber, Allauzen & Waibel 2016, Artetxe, Labaka & Agirre 2018). N-Grams sollen daher als Einstieg für spätere Algorithmen und Anwendungsfälle aus Sicht der Informatik dienen.

3. Statistik

Bei den ausgezählten Kollokationen und N-Grams handelt es sich um absolute Häufigkeiten. Als Gesamtkonstrukt sind sie mathematisch trivial. Die Häufigkeit einzelner Wörter ist jedoch nicht gleichverteilt, sondern folgt dem Zipf-Mandelbrot Gesetz. Dieses Gesetz beschreibt das Verhältnis von Häufigkeiten innerhalb eines beliebig großen Textes oder Korpus'. Als Datengrundlage für die Häufigkeitsanalyse werden die Wörter (oder N-Grams) der Textquelle ausgezählt und nach ihrer Häufigkeit in einer Tabelle geordnet. Das Zipf'sche Gesetz sagt nun, dass das häufigste Wort doppelt so häufig vorkommt, wie das zweit häufigste und dreimal so häufig, wie das Wort auf Position drei. Dieses Verhältnis setzt sich bis zur Hälfte der Häufigkeitstabelle fort. Die untere Hälfte der Häufigkeitstabelle besteht dann nur noch aus $Hapax\ Legomena$, das heißt aus Wörtern, die nur einmal vorkommen (Manning & Schutze 1999). Wird nun die Frequenz jedes Wortes gegen seinen Rang in einem Graphen geplottet und beide Achsen jeweils zur Basis 10 logarithmiert, so entsteht in etwa eine gerade Linie mit der Steigung -1. Mandelbrot erweitert die mathematische Beschreibung, sodass die ersten und die letzten Ränge besser approximiert werden können (Mandelbrot 1965).

Das Zipf-Mandelbrot Gesetz hat zur Folge, dass bei einer zufälligen Kollokation die Wahrscheinlichkeit ihres Auftretens ungleich verteilt ist. Wird die Kollokation eine Position links des Wortes „Versuch" (siehe auch Tabelle 5.8)[78] ausgewertet, so ist die Anzahl an möglichen Partnern zuallererst grammatikalisch begrenzt, was dazu führt, dass hier nur Artikel und Adjektive betrachtet werden. Schon die möglichen bestimmten Artikel „der", „dem" und „den" machen im Bauingenieurwesenkorpus 4,69% aller Wörter aus. Ein passendes Adjektiv findet sich in den häufigsten 100 Wörtern hingegen nicht. Die Wahrscheinlichkeit ist also deutlich höher für eines der 596 Auftreten von „Versuch", einen bestimmten Artikel an der linken Seite anzutreffen als ein bestimmtes Adjektiv. In diesem Fall kann die Wahrscheinlichkeit, dass bei einem beliebigen Auftreten des Wortes „Versuch" einer der genannten Artikel vorausgeht, auf eben jene 4,69% beziffert werden. Allein schon der Artikel „der" ist für 3,35% der Korpusgröße verantwortlich. Jedes andere mögliche Wort hat zwangsläufig eine kleinere Wahrscheinlichkeit. Eine reine Auswertung von Kollokationen auf Basis der absoluten Häufigkeit erscheint daher nicht sinnvoll.

Der MI Score sorgt nun dafür, dass die überproportionale Häufigkeit von grammatikalischen Wörtern ausgeglichen wird. Er setzt bei einer Kollokation aus den Wörtern A und B die Wahrscheinlichkeiten $P(A)$ und $P(B)$ ins Verhältnis zur gemeinsamen Wahrscheinlichkeit $P(AB)$ der Kollokation. Die Wahrscheinlichkeiten für die Wörter A und B ergeben sich aus ihrer Anzahl N_A und N_B geteilt durch die Korpusgröße N. Die Wahrscheinlichkeit der Kollokation ergibt sich aus deren Häufigkeit N_{AB} geteilt durch die Korpusgröße N. Um den MI Score besser interpretieren zu können, wird er zur Basis 2 logarithmiert (Manning & Schutze 1999, Leacock et al. 2014), sodass:

$$\text{MI Score} = \log 2 \frac{P(AB)}{P(A) \cdot P(B)} \tag{5.4}$$

[78] Deklinierte Varianten wie „Versuches" werden hier nicht beachtet.

Der MI Score ist im Gegensatz zu etwa p-Werten bei Signifikanztests nicht konzeptionell, sondern nur mathematisch begrenzt. Für den Fall, dass keine Kollokation gefunden wurde, ist der MI Score minus unendlich. Wird hingegen davon ausgegangen, dass beide Wörter nur einmal vorkommen und dann auch noch als Kollokation, so lässt sich der maximale MI Score anhand von Gleichung 5.9 bestimmen.

$$\text{MI Score} = \log 2 \frac{\frac{1}{N}}{\frac{1}{N} \cdot \frac{1}{N}} \tag{5.5}$$

$$\text{MI Score} = \log 2 \frac{\frac{1}{N}}{\frac{1}{N^2}} \tag{5.6}$$

$$\text{MI Score} = \log 2 \frac{1}{N} \cdot \frac{N^2}{1} \tag{5.7}$$

$$\text{MI Score} = \log 2 \frac{N^2}{N} \tag{5.8}$$

$$\text{MI Score} = \log 2N \tag{5.9}$$

Abbildung 5.17 zeigt das Verhältnis vom MI Score zur Korpusgröße für Korpora von 0 bis 100.000.000 Wörtern. Das Randverhalten für sehr kleine Korpora ist uninteressant, da diese schon aus anderen Gründen nicht für eine Schreibberatung oder linguistische Forschung in Betracht gezogen werden sollten. Für Korpora, die groß genug sind, um analysiert zu werden, aber noch nicht so groß, dass sie nicht mehr von HanConc auf regulärer Hardware verarbeitet werden können, ergeben sich maximale MI Scores von etwa 20 bis 30. Als Grenzwert zwischen zwei zufällig kombinierten Wörtern und einer Kollokation hat sich ein MI Score von drei bewährt (Siyanova & Schmitt 2008).

Gries (2015) schlägt als Alternative zum MI Score Signifikanztests wie t-Tests oder den exakten Test nach Fisher vor. Von einer Implementierung seiner Vorschläge wurde in HanConc aus mehreren Gründen abgesehen: Parametrische Signifikanztests wie der t-Test nehmen bestimmte statistische Verteilungen der Grundgesamtheit an. Wie oben beschrieben, folgt die Häufigkeitsverteilung von Wörtern in einem Text jedoch der Zipf-Mandelbrot Verteilung und nicht einer Normalverteilung wie sie ein t-Test voraussetzt. Somit werden schon die grundlegenden statistischen Annahmen der parametrischen Signifikanztests verletzt. Nichtparametrische Signifikanztests wie der χ^2 Test würden zwar das Problem der verletzten Annahmen heilen (Sheskin 2003), sind jedoch in ihrem Grundgedanken dem MI Score ähnlich, sodass HanConc auf den in der Linguistik weiter verbreiteten MI Score setzt. Für den exakten Test nach Fisher ergibt sich zusätzlich noch das Problem der Berechnung (Raymond & Rousset 1995):

$$P = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}, \tag{5.10}$$

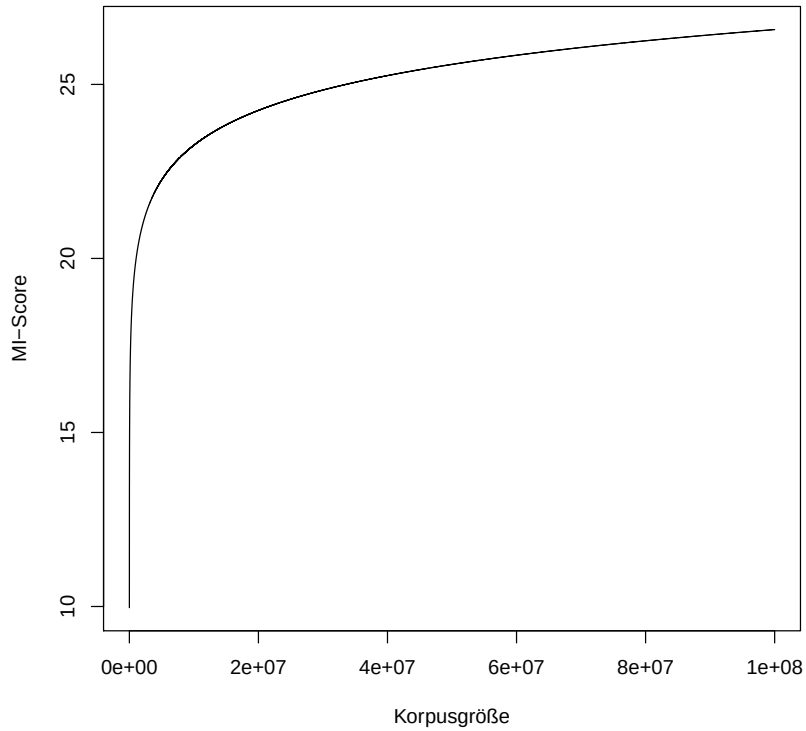


Abbildung 5.17: MI Score im Verhältnis zur Korpusgröße

wobei $\binom{n}{k}$ den Binomialkoeffizienten und ! die Fakultät der entsprechenden Zahl repräsentiert (Zar 1987). Vor allem die Fakultät[79] führt dazu, dass schon bei kleinen Zahlen der RAM nicht ausreicht und HanConc abstürzt.

Insgesamt bietet sich der MI Score als Einstieg in die quantitative Kollokationsanalyse an. Fortgeschrittene Verfahren sollten dennoch auf Basis der vorhandenen Daten und der Erklärungen in diesem und Kapitel 5.3.4 selbst vorgenommen werden.

4. Darstellung im Frontend

Kollokationen, N-Grams und der MI Score werden in HanConcs Frontend jeweils als dreiteilige Tabelle dargestellt. Die Analysen betreffen die Wörter, Lemmata und PoS Tags, welche in Zusammenhang mit dem Suchwort stehen.

Es werden die häufigsten fünf Kollokationen auf den Positionen zwei links bis zwei rechts vom Suchwort dargestellt. Es ist dabei zu beachten, dass es sich, abgesehen vom Zusammenhang zum Suchwort, um unabhängige Elemente handelt. Das bedeutet, dass das häufigste Element auf der Position zwei links nicht zwangsläufig zu dem Element auf der Position eins links passt. Die Häufigkeit des entsprechenden Elements wird in Klammern vermerkt. Für die Wörter wird ebenfalls der auf die zweite Nachkommastelle gerundete MI Score angegeben.

Die N-Grams werden als zwei Tabellen mit je drei Blöcken dargestellt (siehe Abbildung 5.18). Jeweils auf Wort-, Lemma- und PoS Tag-Ebene werden die fünf häufigsten Bi- und Tri-

[79] Fakultät bedeutet, dass alle ganzen positiven Zahlen einschließlich der beschriebenen miteinander multipliziert werden. Für $5!$ bedeutet dies: $0 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$.

Grams präsentiert. Im Falle der Bi-Grams enthält die Tabelle zwei Spalten, wobei in der ersten das Suchwort rechts und in der zweiten Spalte links steht. Bei Tri-Grams wird in der Mitte eine weitere Spalte eingefügt, in der das Suchwort ebenfalls in der Mitte steht. Die Häufigkeit des N-Grams wird durch die vorgestellte Zahl markiert.

Frequency and n-Gram

2-grams (Words)

16	der Methode	31	Methode der
16	die Methode	16	Methode zur
12	diese Methode	7	Methode auf
9	eine Methode	5	Methode ,
5	dieser Methode	3	Methode des

2-grams (Lemmata)

18	die Methode	31	methode der
16	der Methode	16	methode zur
13	diese Methode	7	methode auf
9	eine Methode	5	methode ,
5	dieser Methode	3	methode des

2-grams (POSta)

44	ART Methode	34	Methode ART
23	ADJA Methode	17	Methode APPRART
18	PDAT Methode	13	Methode APPR
5	$[Methode	5	Methode $.
2	PDAT ART Methode Methode	4	Methode VVPP

3-grams (Words)

9	mit der Methode	14	die Methode der	7	Methode der komplexen
5	durch die Methode	12	der Methode der	5	Methode der Finiten
4	wird diese Methode	5	diese Methode auf	4	Methode der Semi-Diskretisierung
3	' beschriebene Methode	5	eine Methode zur	4	Methode der stueckweise
3	' beschriebenen Methode	2	' Methode zur	3	Methode der kleinsten

3-grams (Lemmata)

9	mit der Methode	14	die Methode der	7	methode der komplexen
6	durch die Methode	12	der Methode der	5	methode der finiten
4	wird diese Methode	5	diese Methode auf	4	methode der semi-diskretisierung
3	' beschriebene Methode	5	eine Methode zur	4	methode der stueckweise
3	' beschriebenen Methode	2	' Methode zur	3	methode der kleinsten

3-grams (POSta)

15	APPR ART Methode	29	ART Methode ART	18	Methode ART ADJA
8	$[ADJA Methode	7	ADJA Methode APPRART	14	Methode APPRART NN
8	NN ART Methode	7	PDAT Methode APPR	13	Methode ART NN
7	ART ADJA Methode	6	ART Methode APPRART	5	Methode APPR ART
7	VAFIN ART Methode	3	ART Methode APPR	3	Methode APPR NN

Abbildung 5.18: Beispielhafte Darstellung von N-Grams in HanConc

5. Nutzen in der Schreibberatung

Kollokationen und N-Grams sollten neben der Frequenz und den KWIC die am häufigsten genutzten Funktionen sein. Sie erlauben es, die KWIC quantitativ zu aggregieren und zeigen das Suchwort in seiner näheren Umgebung. Vor allem vor dem linguistischen Hintergrund von im Text nahe beieinander stehenden semantisch zusammengehörigen Wortgruppen stellen Kollokationen und N-Grams eine wichtige Funktion für die Schreibberatung dar. Venohr schreibt in diesem Zusammenhang, dass vor allem Studierende, deren Muttersprache nicht Deutsch ist, auf Kollokationen zurückgreifen, da im Bewusstsein ihres limitierten Vokabulars das „Auffinden einer passenden Kollokation [...] zu [ihren] lernersprachlichen Strategien gehört" (Venohr & Neis 2013, 8). Einig & Menne-El Sawy (2012) weisen darauf hin, dass Kollokationen häufig „misslingen" (388) und besonders unterrichtet werden müssten. Aus diesen Erkenntnissen lässt sich positiv formulieren, dass Kollokationen die sprachliche Qualität erheblich steigern können. Ulmi, Bürki, Marti & Verhein-Jarren (2017) weisen darauf hin, dass Kollokationen, wie oben beschrieben, unterschiedlich variabel sind. Während einige Wörter frei zu Kollokationen kombiniert werden können, handelt es sich bei anderen um feststehende Konstruktionen. Als pädagogische Schlussfolgerung ziehen sie daraus, das Kollokationen

als Zeichen fortgeschrittenen Sprachgebrauchs trainiert und auswendig gelernt werden müssen (Ulmi, Bürki, Marti & Verhein-Jarren 2017).

Laut Steinhoff (2010) können Kollokationen vor allem dazu eingesetzt werden, den eigenen Text an einen anvisierten wissenschaftlichen Sprachstil anzupassen. Kollokationen werden hierbei als über die reine Lexik hinausgehende Voraussetzung für wissenschaftliches Schreiben gesehen. Erst der richtige Gebrauch entsprechend der jeweiligen „Domänentypik" (81) erlaubt es, angemessene wissenschaftliche Texte zu schreiben (Steinhoff 2010).

Jafarpour & Sharifi (2012) untersuchen den Effekt von explizitem Unterrichten zur Korrektur von fehlerhaften Kollokationen. Iranische Studierende der englischen Übersetzungswissenschaften dienen hierbei als Versuchspersonen. Ihnen werden als Multiple-Choice Test häufig falsch verwendete Kollokationen vorgelegt, aus denen sie die richtige Kollokation wählen sollen. Im Anschluss werden die korrekten Formen der häufigsten fehlerhaften Kollokationen explizit unterrichtet und der Test nach einigen Monaten wiederholt. Die Studierenden werden entsprechend ihrer Fähigkeiten in drei Gruppen und die Gesamtpopulation in eine Test- und eine Vergleichsgruppe eingeteilt. Die Studie zeigt, dass vor allem die beiden Gruppen mit der höheren Sprachkompetenz durch den Unterricht signifikant besser werden. Laut der Literaturauswertung in Jafarpour & Sharifi (2012) gibt es Evidenzen für als auch gegen die These, dass durch Unterricht die Fähigkeit, Kollokationen richtig einzusetzen, verbessert werden kann.

Für diese Arbeit ergeben sich aus den exemplarisch ausgewählten Studien zwei Thesen, die von den Studienautor_innen als gegeben angesehen und hier diskutiert werden müssen: Die Autor_innen gehen davon aus, dass es richtige und falsche Kollokationen gibt, welche trennscharf unterschieden werden können. Außerdem wird angenommen, dass die unterrichtende Person diese Unterscheidung vornehmen kann. Andere ähnlich gelagerte Publikationen basieren auf den gleichen Annahmen (Chang, Chang, Chen & Liou 2008, Tschichold et al. 2003, Diab 2015, Bower & Kawaguchi 2011, Dawood 2014, Luo & Liao 2015, Chan & Liou 2005, Han & Hyland 2015, Cuéllar 2013).

Es muss davon ausgegangen werden, dass die Standardisierung von Kollokationen mit der Spezialisierung ihrer Elemente abnimmt. Dies bedeutet, dass allgemeinsprachliche Kollokationen durchaus soweit kanonisiert sind, dass sie etwa in Wörterbüchern auftauchen. Die korrekte Verwendung von Kollokationen in der Wissenschaftssprache ist jedoch gegebenenfalls nur aus den abgegrenzten Fachbereichen zu beurteilen. Im Falle der in HanConc hinterlegten Texte muss davon ausgegangen werden, wie in Kapitel 4.4.3 gezeigt, dass selbst innerhalb einzelner Fakultäten die sprachliche Spezialisierung so weit fortgeschritten ist, dass Kollokationen kleinteiliger beurteilt werden müssen. Deshalb muss davon ausgegangen werden, dass außerhalb der Grenzen der jeweiligen Fachdisziplin die reine erstsprachliche Kompetenz der Schreibberater_in nicht ausreichend ist, um die Angemessenheit von Kollokationen zu beurteilen. Aus diesem Grund muss die Fähigkeit von Schreibberater_innen, Kollokation zu unterrichten, kritisch hinterfragt werden. Abgesehen von wenigen Ausnahmen[80], ist nicht davon auszugehen, dass Nachschlagewerke für fachspezifisches Vokabular und für die dazugehörigen Kollokatio-

[80] Etwa Schroth-Wiechert (2011)

nen zur Verfügung stehen. Wird zusätzlich davon ausgegangen, dass keine Deckung des akademischen Hintergrunds zwischen Schreibberater_in und Student_in vorliegt, ist ein Unterrichten, wie es Jafarpour & Sharifi (2012) vorsehen, nicht möglich.

Da also weder abschließend geklärt werden kann, welche Kollokationen im Sinne der oben genannten Publikationen richtig sind und Schreibberater_innen die Qualität der Kollokationen ebenfalls nicht immer beurteilen können, so muss auf ein deskriptives Verfahren zur Analyse von Kollokationen zurückgegriffen werden. HanConc beispielsweise kann genutzt werden, um im Sinne der Frequenz häufige und im Sinne des MI Scores starke Kollokationen zu identifizieren und diese dann mit den einschlägig bekannten Verfahren des „Computer Assisted Language Learning" (CALL) zu unterrichten (Daskalovska 2015, Kohn 2009, McEnery, Baker & Wilson 1995, Chang & Chang 2004, Nerbonne, Dokter & Smit 1998, Belz 2004, Kennedy & Miceli 2010, Boulton 2010, Heift & Schulze 2007, Sha 2010).

Leacock et al. (2014) verwenden einen ähnlichen Ansatz. Sie nutzen unter anderem den MI Score, um relevante Kollokationen in muttersprachlichen Texten zu identifizieren. Anstatt diese jedoch Lehrenden und Studierenden einfach nur zu präsentieren, wird ein automatisiertes System entwickelt, um falsche Kollokationen in den Texten der Lernenden zu bestimmen und die entsprechend richtige Kollokation als Berichtigung anzuzeigen. Richtig und falsch wird in diesem Fall auf Basis von Signifikanztests unterschieden. Schreibberatungen könnten grundsätzlich von diesem Ansatz auch profitieren. Allerdings ergeben sich für das Schreibzentrum der LUH, für welches HanConc und die dazugehörigen Korpora entwickelt wurden, einige Hindernisse: Studierende am Schreibzentrum der LUH werden zwar im Prozess des Schreibens unterstützt, allerdings werden dort keine fertigen Arbeiten korrigiert. Außerdem ist die Datengrundlage wahrscheinlich zu gering, um valide Signifikanztests zu Kollokationen durchzuführen. HanConc soll deshalb vor allem dazu dienen, Studierenden mögliche sinnvolle Kollokationen anzubieten, anstatt sie relevante Kollokationen durch ein Feedbacksystem erraten zu lassen.

Mit Verweis auf die unterschiedlichen linguistischen Definitionen von Kollokationen muss beachtet werden, dass HanConc Kollokationen als Kookkurrenzen definiert. Dies bedeutet, dass zusammenhängende Mehrwortkombinationen, die im Text nicht direkt hintereinander stehen, von HanConc nicht als Kollokation erkannt werden. Es müssen komplexere Mehrwortsuchen eingesetzt werden, um solche Kollokationen ausfindig zu machen und zu analysieren. Allerdings bietet der Kollokationsergebnistyp einen Startpunkt für weitere Analysen.

5.3.8.4 Position in der Wortliste

0. Einleitung

Bei der Integration in HanConc wird jedes Korpus ausgezählt und je zwei Listen erstellt, welche jeweils ein Wort pro Zeile und dessen Frequenz im Korpus enthalten. Beide Listen unterscheiden sich dadurch, dass erstere Liste Worte inklusive Groß- und Kleinschreibung, konjugierten

und deklinierten Formen enthält, während die zweite aus Lemmata besteht[81]. Im Frontend wird die Position in der Wortliste als einfacher Ausschnitt aus einer Rangliste präsentiert. Allerdings sind die statistischen Auswirkungen immens. Vor allem in Kombination mit der Frequenz und der Statistikfunktion ermöglicht die Feststellung der Position in der Wortliste ein genaues Bild über die Verteilung und die Funktion einzelner Wörter in einem Korpus.

1. Input

Die Wortlisten werden vorberechnet und als DataFrames in den jeweiligen Korpusobjekten abgelegt, wobei es sich hierbei um R Listen handelt. HanConc durchsucht die Listen nach dem (Haupt-)Suchwort und gibt zusätzlich noch die umliegenden Zeilen zurück. Außerdem wird berechnet, wie viele Wörter nur einmal vorkommen ($Hapax\ Legonmena$) und wie lang die Liste insgesamt ist.

2. Linguistik & 3. Statistik

Die Statistikfunktion (siehe auch Kapitel 5.3.8.6) überprüft die beobachtete Verteilung der Anzahl an Suchergebnissen pro Text auf mehrere Standardverteilungen. All diese Verteilungen haben gemein, dass sie sich um einen Mittelwert orientieren und relativ schmale Ränder haben. Bei einer Standardnormalverteilung etwa sind Mittelwert und Median gleich und Maximum und Minimum sowie 1. Quartil und 3. Quartil jeweils gleich weit vom Mittelwert entfernt. Wie Briscoe oder Newman zeigen, wird bei vielen alltäglichen Beobachtungen wie Körpergröße und -gewicht intuitiv von einer Normalverteilung ausgegangen. Anhand des FElt Korpus soll nun gezeigt werden, dass die Wortanzahl in einem Text einer grundsätzlich anderen Verteilung folgt und somit die Notwendigkeit einer Wortlistenfunktion motiviert.

Linguistische Grundlage für die Wortlistenfunktion ist dabei die Idee von Types und Tokens (Bußmann & Lauffer 2008). Jedes unterschiedliche Wort entspricht einem Type. Jedes Auftreten dieses Wortes entspricht seinem Token. Die bisherigen drei Sätze oder 26 Wörter dieses Absatzes bestehen beispielsweise aus 23 Types mit je einem Token[82]. „die", „jedes" und „entspricht" kommen doppelt vor und haben deshalb je zwei Token. Es ist grundsätzlich so, dass grammatikalische Notwendigkeiten wie Artikel, Präpositionen oder Pronomen dafür sorgen, dass sich eine große Anzahl an Tokens auf wenige Types konzentriert. Wird ein Korpus so aufbereitet, dass eine Tabelle mit Types in der ersten Spalte und der jeweiligen Anzahl an Tokens in der zweiten Spalte entsteht, ergibt dies eine Wortliste.

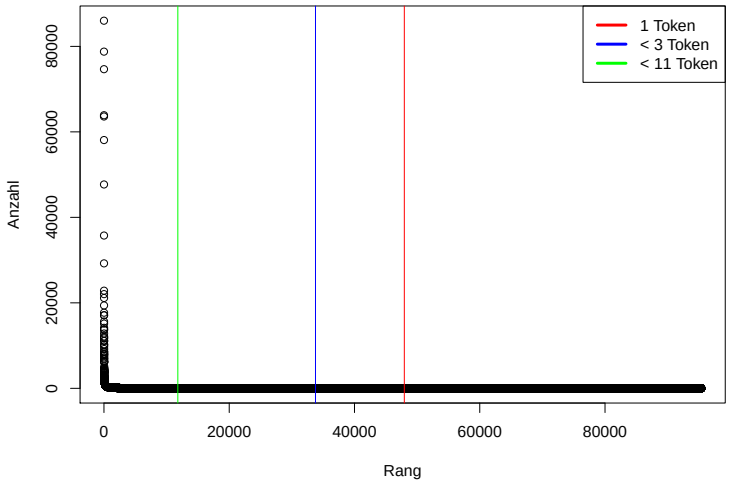
Abbildung 5.19 zeigt die Auswertung der Wortliste des FElt Korpus. In (a) sind die 20 häufigsten Worte bzw. Satzzeichen aufgelistet, während in (b) die Anzahl an Types pro Token gegenüber dem Rang in der Wortliste geplottet sind. In beiden Fällen ist eine rasche Abnahme der Tokens pro Type festzustellen. Insgesamt umfasst das FElt Korpus 95.380 individuelle Wörter. Von diesen 95.380 Types haben 47.426 oder 49,72% nur einen einzigen Token. Weitere 14.180 Types oder 14,87% des Korpus haben nur zwei Tokens. Wird die Anzahl an Tokens auf

[81] Es ist zu beachten, dass für das Deutsche zum jetzigen Stand kein Lemmatisierer in ausreichend guter Qualität zur Verfügung steht. Anstatt der Lemmata wurden die Worte aus der ersten Liste in die zweite Liste übertragen und dabei die Großbuchstaben zu Kleinbuchstaben konvertiert, um etwa den Einfluss von Satzanfängen auszugleichen. Die zweite Liste wurde vor allem dafür angelegt, dass ein entsprechender Platz zur Verfügung steht, sobald ein guter Lemmatisierer verfügbar ist.

[82] Der Quellenverweis wird hier nicht mitgerechnet.

(a) 20 häufigste Wörter und Satzzeichen im FElt Korpus

	Type	Anzahl Token
1	.	86.011
2	,	78.776
3	der	74.665
4	die	58.090
5	NUMBER	47.698
6	und	35.752
7	in	29.239
8	von	22.848
9	des	19.392
10	den	17.699
11	ist	17.102
12	werden	17.090
13	zu	15.633
14	mit	15.151
15	auf	14.160
16	eine	13.817
17	fuer	13.651
18	Die	12.839
19	wird	12.150
20	im	11.898



(b) Rang/Anzahl Plot für das FElt Korpus mit markierten Tokengrenzen

Abbildung 5.19: Tabellarische und grafische Darstellung der Type/Token Verteilung im FElt Korpus

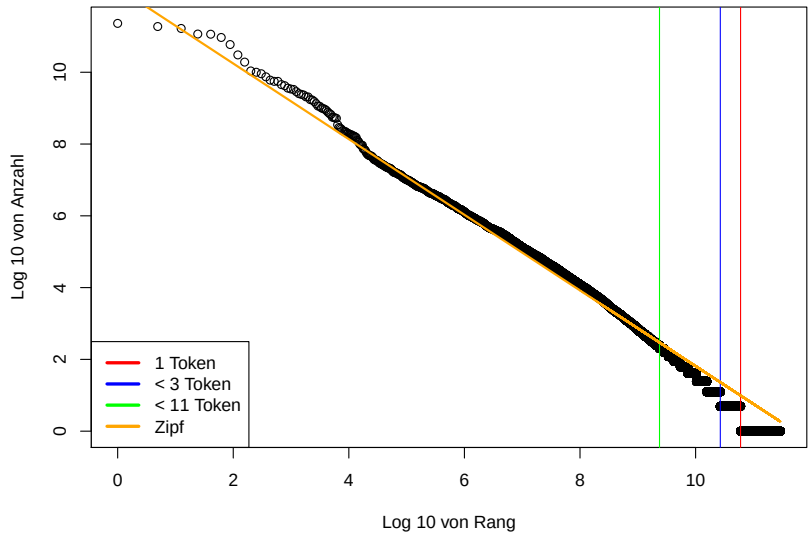


Abbildung 5.20: Abbildung 5.19b mit logarithmierten Achsen und gefitteter Zipfverteilung

mindestens 10 pro Type erhöht, erfüllen 88,63% des Korpus' diese Anforderung nicht.

Neben der Verteilung von Types und Tokens in natürlicher Sprache gibt es noch weitere Phänomene, die sich ebenso verhalten. Briscoe (2007) führt etwa die Einwohnerzahl von Städten oder Webseitenaufrufe an, sowie die Verteilung von Reichtum in Gesellschaften. In all diesen Fällen konzentriert sich der Großteil der zur Verfügung stehenden Mittel auf die obersten Prozentränge. Das Jahresgutachten 2009/2010 des Sachverständigenrates der deutschen Wirtschaft stellt etwa in Tabelle 40 die Verteilung von Einkommen nach Dezentilen dar: Die obersten 10% der Einkommensverteilung konzentrieren 31,2% des Gesamteinkommens auf sich, während die untersten 50% nur 17,3% des Gesamteinkommens erhalten (Sachverständigenrat zur Begutachtung der Gesamtwirtschaftlichen Entwicklung 2009).

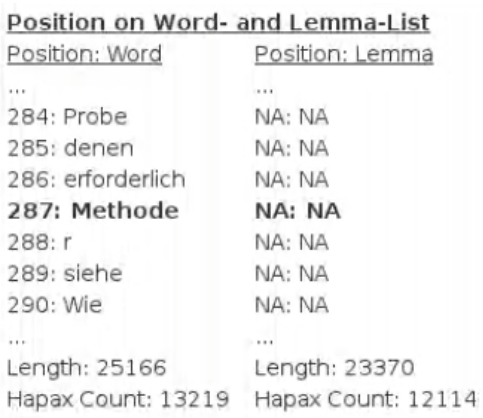
Bei diesem Verhalten handelt es sich um eine Verteilung aus der Familie der Power Laws, welche auf den Studien von Pareto (1964) basieren. Obwohl die Paretoverteilung vor allem für wirtschaftliche Zusammenhänge gedacht war, kann sie auch auf linguistische Phänomene angewandt werden. Zipf erweiterte die Paretoverteilung zum Zipfschen Gesetz (Zipf 1949), welches von Mandelbrot (1965) um einen weiteren Parameter zum Zipf-Mandelbrot Gesetz ergänzt wurde.

Abbildung 5.20 zeigt eine Abwandlung von Abbildung 5.19b. In dieser Variante sind beide Achsen zur Basis 10 logarithmiert. Der hierdurch entstehende Graph erinnert nicht mehr an ein liegendes L wie in der vorherigen Abbildung, sondern an eine annähernde Gerade mit negativer Steigung. Dem zipfschen Gesetz folgend, wurde die tatsächliche Verteilung gefittet und mit den ermittelten Parametern eine theoretische Verteilung hinzugeplottet.

Für HanConc bedeuten die oben beschriebenen Verteilungen, dass nachfolgende Funktionen nur bei einer ausreichend großen Datenmenge robuste Ergebnisse liefern können.

4. Darstellung im Frontend

Neben der offensichtlichen Position auf der Wort- und Lemmaliste wird im Frontend auch die



Position on Word- and Lemma-List	
Position: Word	Position: Lemma
...	...
284: Probe	NA: NA
285: denen	NA: NA
286: erforderlich	NA: NA
287: Methode	**NA: NA**
288: r	NA: NA
289: siehe	NA: NA
290: Wie	NA: NA
...	...
Length: 25166	Length: 23370
Hapax Count: 13219	Hapax Count: 12114

Abbildung 5.21: Beispielergebnis für die Funktion „Position in der Wortliste”

Anzahl an Types und $Hapax\ Legomena$, d.h. die Anzahl an Types mit nur einem Token, dargestellt. Abbildung 5.21 zeigt am Beispiel von „Methode” die Ergebnisse der Wortlistenfunktion im FElt Korpus. Neben der eigentlichen Position werden auch die drei höheren bzw. niedrigeren Positionen zurückgegeben. Außerdem wird in den letzten beiden Zeilen die Anzahl an Types und die Anzahl an Types mit nur einem Token angegeben. Da bei dem hinterlegten Korpus keine Lemmata vorhanden sind, enthält die entsprechende Ergebnisspalte keine Werte.

5. Nutzen in der Schreibberatung

Der Nutzen für die Schreibberatung ergibt sich aus den statistischen Zusammenhängen. Wenn davon ausgegangen wird, dass die Verteilung von Types und Tokens in einem Korpus einer Zipf-Mandelbrot Verteilung folgt und es einer ausreichend großen Menge an Treffern bedarf, um komplexere statistische Verfahren durchzuführen, so ist die Position in der Wortliste zusammen mit der Frequenzfunktion die zentrale Einstiegsfunktion für alle Analysen, die über die KWIC hinausgehen.

Die Frequenz eines Suchbegriffs benötigt als weitere Kennzahlen die Korpusgröße und -beschaffenheit. Wird Sprache als Bernoulli Funktion gesehen, bei der jedes Wort in einem bestimmten Kontext eine dezidierte Wahrscheinlichkeit des Auftretens hat, so steigt mit der Größe des Korpus' die Wahrscheinlichkeit, dass das Wort tatsächlich auftritt bzw. häufiger auftritt. Die Fokussierung auf die Korpusgröße im Sinne einer Gesamtwortanzahl erscheint angesichts der Verschiebung des Anteils zugunsten von grammatikalischen Wörtern bei wachsender Korpusgröße wenig sinnvoll.

Die obige statistische Beschreibung hat gezeigt, dass sich die Verteilung der Häufigkeiten von Wörtern innerhalb eines Korpus' folgendermaßen gestaltet: Die häufigsten 10% der Types machen einen Großteil der Tokens aus, 50% der Types kommen nur ein einziges Mal vor, während die übrigen 40% mit einer mittleren Frequenz erwartet werden können. Anhand der Verteilung und der drei Informationen (Position auf der Wortliste, Länge der Wortliste und Anzahl an $Hapax\ Legomena$), die die Wortlistenfunktion zurückgibt, kann die Verbreitung der Verwendung des Suchwortes abgeschätzt werden.

Für die Schreibberatung ergibt sich aus der Wortlistenfunktion der notwendige Kontext, um die Frequenz und die Anzahl an Kollokationen und N-Grams einschätzen zu können. Erst

dadurch kann analysiert werden, ob es sich bei den Suchergebnissen um statistische und linguistische Artefakte handelt.

5.3.8.5 Wortassoziationen auf Basis einer Latent Semantic Analysis

0. Einleitung

Außerhalb von Schreibberatung und Linguistik gibt es verschiedene Verfahren und Algorithmen, die es Nutzer_innen erlauben, einzelne Elemente in einer großen Anzahl von Elementen zu finden, ohne sie vorher explizit zu kennen. Vor allem der datengetriebene Verkauf von Produkten und Dienstleistungen hat Methoden entwickelt, mehr anzubieten und schlussendlich zu verkaufen, als es die initiale Intention der potentiellen Kund_innen war. HanConc und Schreibberatung könnten von der Verwendung solcher Algorithmen und Verfahren profitieren.

In Verkaufssituationen sollen Kund_innen davon überzeugt werden, möglichst viele zusätzliche Produkte und Dienstleistungen zu erwerben. Automobilhersteller verkaufen beispielsweise Versicherungen oder Wartungsverträge zu ihren Autos. Diese Herangehensweise zur Umsatzsteigerung wird aus Marketingsicht „Cross-Selling" genannt (Homburg & Krohmer 2009, Homburg & Schäfer 2002). Im stationären Handel ist die Anzahl an Produkten und Dienstleistungen zeitlich und räumlich begrenzt und damit auch die Möglichkeiten für Cross-Selling. In Kundengesprächen können kaum mehr als eine handvoll Zusatzdienstleistungen vorgestellt werden und Supermärkte haben nur begrenzt Platz, um sinnvolle Produktkombinationen nebeneinander zu platzieren. Onlinehändler hingegen sind nicht durch Regalmeter begrenzt und durch Suchalgorithmen und Datenbanken kann eine beliebig große Anzahl an Dienstleistungen und Produkten vorgehalten werden.

Suchalgorithmen für Onlineshops müssen in zwei Richtungen optimiert werden: Weil keine menschliche Verkäufer_in zur Verfügung steht, die auf Basis der Beschreibung eines Produktes durch potentielle Kund_innen realisieren kann, welches Produkt gemeint ist, kann eine Fuzzy Search eingesetzt werden (Ji, Li, Li & Feng 2009). Fuzzy Search Algorithmen wie etwa Levenshtein Distanzen versuchen, auf Basis der eingegebenen Buchstaben, mögliche ähnliche Suchbegriffe zu finden und anzuzeigen. So können zum Beispiel Rechtschreibfehler im Suchbegriff ausgeglichen werden (Hirschberg 1997, Van der Loo 2014, Navarro 2001, Shang & Merrettal 1996). Zusätzlich sollen durch Warenkorbanalysen sinnvolle Produktkombinationen ermittelt und der Kund_in vorgeschlagen werden. Sucht eine Kund_in nach einer Bohrmaschine, so sollen ihr auch die passenden Bohrköpfe angezeigt werden. Diese Warenkorbanalysen basieren auf dem Kaufverhalten früherer Kund_innen (Agrawal, Srikant et al. 1994, Srikant & Agrawal 1995, Hsueh & Kuo 2017, Yuan 2017, Fan, Wang, Wu & Xu 2015, Yoshimura, Sobolevsky, Bautista Hobin, Ratti & Blat 2018).

Die Funktionen von HanConc produzieren bisher nur genau die Ergebnisse, die von den Nutzer_innen explizit angefordert werden. Sollen auch Ergebnisse angezeigt werden, die sinnvoll über die Erwartungen hinausgehen, müssten dafür komplexere Algorithmen verwendet werden. Für eine Fuzzy Search bräuchte es Suchalgorithmen, die über eine einfache Schleife hinausgehen. Da der Fokus bei der Programmierung von HanConc auf Einfachheit und Mani-

pulierbarkeit liegt, wird auf Fuzzy Search Algorithmen demnach verzichtet.

Um beispielsweise das oben beschriebene Verfahren wie die Warenkorbanalyse in HanConc zu implementieren, bräuchte es neben den Suchalgorithmen auch eine große Datenbank an bereits durchgeführten Suchen mit eindeutigen Zeitstempeln und Benutzermerkmalen. Neben praktischen Problemen wie einer zu geringen Anzahl an Schreibberatungen mit Einsatz von HanConc, zu wenigen Studierenden in der Schreibberatung für eine ausreichende Datengenerierung und der Notwendigkeit für eine zentralisierte Datenhaltung sprechen vor allem Datenschutz und Designentscheidungen gegen eine solche Datenbank. HanConc ist bewusst so gestaltet worden, dass es möglichst isoliert und dezentral betrieben werden kann. Außerdem würde das Speichern von Benutzerdaten dazu führen, dass nationale und europäische Datenschutzgesetze sowie die Datenschutzrichtlinien der Institutionen, in denen HanConc eingesetzt werden soll, beachtet werden müssten.

HanConc setzt deshalb auf eine Lösung, die vollkommen autark und ahistorisch funktioniert. Mit einer Term-Dokumenten Matrix (TDM) und einer Latent Semantic Analysis (LSA) können semantisch ähnliche Wörter allein auf Basis der Korpusdaten zurückgegeben werden. Diese semantisch ähnlichen Wörter können wiederum in einer Schreibberatung verwendet werden, um weitere Suchen durchzuführen, bisherige Suchen zu verfeinern oder das Vokabular der Studierenden zu erweitern. Ebenso eignet sich diese Lösung, um weitere und komplexere Text-Mining Algorithmen zu implementieren.

1. Input

Wie in Kapitel 5.3.5 bereits ausgeführt[83] dienen TDM als Basis für eine LSA. Für Analysen mittels der LSA wird jedes Korpus als eigenständige TDM aufbereitet. Für jeden Begriff/Type wird eine Zeile und für jedes Dokument eine Spalte in der jeweiligen TDM angelegt. Jede Zelle enthält bei einer ungewichteten Matrix die Anzahl des jeweiligen Begriffs im entsprechenden Dokument. Bei einer LSA werden je nach Einsatzzweck einzelne Zeilen oder Spalten der TDM miteinander verglichen.

Die Erstellung einer solchen Textmatrix benötigt einige Aufbereitungsschritte. Als Grundlage dienen einzelne TXT Dateien, welche nach Korpus in Ordner sortiert sind. Da HanConc das LSA R Paket nutzt, kommt die Textaufbereitung mit wenigen Zeilen Quellcode aus. Die Pfade zu den Daten werden in einen Vektor eingelesen. Es ist zu beachten, dass der letzte Ordnername über den abschließenden Dateinamen der Textmatrizen entscheidet. Der Vektor mit den Dateipfaden wird an eine Schleife übergeben, welche die Aufbereitungsschritte je Korpus durchführt.

In einem ersten Schritt werden die Texte eingelesen und gesäubert. Das LSA Paket bietet das Entfernen von Zahlen an, was jedoch im Vorfeld bereits geschehen ist. Die einzige weitere notwendige Option ist das Entfernen der Stopwords, was bedeutet, dass ein Großteil des funktionalen Vokabulars wie Artikel entfernt werden. Sollen Texte in einer anderen Sprache als Deutsch eingelesen werden, so ist die Option entsprechend anzupassen. Die minimale Anzahl an notwendigen Buchstaben je Wort ist auf zwei festgesetzt. Außerdem kann eine Gewich-

[83] Kapitel 5.3.5 beleuchtet die Textaufbereitungsschritte aus methodischer Sicht. An dieser Stelle soll jedoch allein auf die Programmierung eingegangen werden.

tung an dieser Stelle vorgenommen werden. Die so erstellte Textmatrix ist weder gewichtet noch durch eine Singular Value Decomposition (SVD) reduziert. Diese Schritte erfolgen durch den Aufruf der LSA Funktion. Die Parameter für die SVD werden durch die im LSA Paket implementierte Funktion bestimmt. Das entstehende R Objekt wird zurück in eine Textmatrix überführt und entsprechend des Korpusnamens auf der Festplatte gespeichert.

2. Linguistik

Grundlage jeder LSA ist eine TDM oder, wenn diese transponiert wird, eine Dokumenten-Term Matrix (DTM). Ziel einer LSA ist es, die Ähnlichkeit der einzelnen Elemente in Zeilen- bzw. Spaltenrichtung zu untersuchen. Die Ausgestaltung der einzelnen Elemente ist dabei vom Anwendungsfall abhängig. Der Begriff „Term" ist also nicht nur auf einzelne Wörter beschränkt, sondern kann ebenfalls Wortteile oder N-Grams beinhalten. Ebenso muss sich ein „Dokument" nicht nur auf einen gesamten Text beziehen, sondern kann auch für einzelne Passagen oder gar einzelne Sätze stehen. CohMetrix etwa setzt eine LSA zur Überprüfung von Kohäsion zwischen einzelnen Sätzen ein (McNamara et al. 2014). Zusätzlich kann sie auch zum unüberwachten Modellieren von Themen innerhalb eine Korpus' eingesetzt werden (Chang, Gerrish, Wang, Boyd-Graber & Blei 2009).

Die LSA oder das Latent Semantic Indexing (LSI)[84] wurde von Scott Deerwester, Susan Domais und Thomas Landauer in den späten 1980ern entwickelt. Die Forschungsgruppe arbeitete an der automatischen Extraktion von Informationen aus Texten, wobei sich ein Problem mit Synonymen ergab. 1987 konnten Furnas, Landauer, Gomez & Dumais zeigen, dass von verschiedenen Personen in den meisten Fällen unterschiedliche Wörter für das gleiche Referenzobjekt verwendet werden. Dies führt dazu, dass eine Informationsextraktion auf Wortebene wenig erfolgsverspechend ist. In Kombination mit dem Prinzip von Firth „You shall know a word by the company it keeps" (Evert 2005) lässt sich das Problem durch die Inklusion des latenten Kontextes eines Wortes lösen. Zwar kann für ein einzelnes Wort die Bedeutung nicht genau bestimmt werden, jedoch führt die Kombination aus vielen vage zu bestimmenden Wortbedeutungen zu einer relativ genauen Festlegung (Deerwester, Dumais, Furnas, Landauer & Harshman 1990). Es ist hierbei zu bedenken, dass die Genauigkeit dieser Festlegung mit der Größe des Korpus' steigt, da so mehr Kontexte betrachtet werden können, in denen die einzelnen Wörter vorkommen. Dementsprechend verlagert sich die Diskussion der LSA von einem linguistischen zu einem statistischen Problem (siehe Unterpunkt 3 dieses Kapitels).

Neben den oben erwähnten Möglichkeiten wird eine LSA unter anderem auch zur automatisierten Textbewertung (Sukkarieh, Pulman & Raikes 2003), Diskursanalyse (Hempelmann, Dufty, McCarthy, Graesser, Cai & McNamara 2005) oder Synonymidentifikation (Turney 2001) eingesetzt. Vor allem bei der computergestützten Bewertung von großen Mengen an standardisierten Essays wird eine LSA verwendet, um die Ähnlichkeit zum Erwartungshorizont zu überprüfen und damit einzelne Essays automatisiert zu bewerten (Burstein & Chodorow 2010). Neuere Herangehensweisen wie Word Embeddings oder Word-to-Vector basieren auf neuronalen Netzwerken und versprechen eine höhere Genauigkeit und Reliabilität (Mikolov, Corrado,

[84] Es handelt sich um das gleiche Verfahren, welches je nach akademischer Ausrichtung des Autors entweder LSA oder LSI genannt wird (Chang et al. 2009, 2).

Chen & Dean 2013). Allerdings wird an dieser Stelle auf eine eingehendere Diskussion und Umsetzung dieser Möglichkeiten verzichtet, da das zur Verfügung stehende Korpus zu klein ist und das Training eines neuronalen Netzwerkes im Gegensatz zu einer LSA die Hardwareanforderungen und den zeitlichen Aufwand für die Textaufbereitung massiv beeinträchtigen würden.

3. Statistik

Eine LSA basiert auf einer Matrix *M* mit den Dimensionen *t* und *w*, wobei *t* den Texten und *w* den enthaltenen Wörtern entspricht. In ihrer Ursprungsform entspricht diese Matrix einer ungewichteten TDM. Der Nachteil einer ungewichteten Matrix besteht jedoch in ihrer hohen Abhängigkeit von der Länge der zu analysierenden Texte. Eine Gewichtung kann etwa durch relative Frequenzen ausgedrückt werden, sodass für das Wort *i* in Text *j* gilt:

$$M_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{n} t_{k,j}} \tag{5.11}$$

Diese Art der Gewichtung berücksichtigt nur die Korpusgröße. Alternative Gewichtungen erlauben es, auch das grundsätzliche Thema des Korpus' zu betrachten. In diesem Fall wird für das Wort w_i in Text t_i die Anzahl an Wort w_i mit der Korpusgröße multipliziert und durch die Anzahl des Wortes w_i im Korpus dividiert, sodass:

$$M_{i,j} = \frac{w_{i,j} \cdot \sum_{k=1}^{n} t_{k,j}}{\sum_{k=1}^{n} w_{i,k}} \tag{5.12}$$

Somit wird sichergestellt, dass neben Wörtern, die häufig im betrachteten Text vorkommen, auch solche Wörter heruntergewichtet werden, die häufig im gesamten Korpus verwendet werden. Für HanConc führt dies dazu, dass sowohl grammatikalische Wörter als auch allgemeine Wissenschaftssprache weniger Einfluss erhalten und fachspezifische Unterschiede mehr zum Tragen kommen. Diese Art der Gewichtung wird Term Frequency-Inverse Document Frequency (TF-IDF) genannt (Francis & Flynn 2010) und wird bei der Implementierung der LSA in R definiert als:

$$w\ one - hot_{i,j} = \begin{cases} 1; & w_{i,j} > 0 \\ 0; & w_{i,j} = 0 \end{cases} \tag{5.13}$$

$$M_{i,j} = w_{i,j} \cdot (1 + log_2(\frac{N_t}{\sum_{k=1}^{n} w\ one - hot_{i,k}})) \tag{5.14}$$

Mit dieser Methode wird die Textanzahl durch die Anzahl an Texten dividiert, die das jeweilige Wort mindestens einmal enthalten. Dieser Term wird zur Basis 2 logarithmiert, um die Funktion zu glätten. Zu diesem Teil wird eins multipliziert, um sicherzugehen, dass auch wenn in jedem Text das betrachtete Wort vorkommt, die Einzelfrequenzen berücksichtigt werden können. Andernfalls würde jedes Wort, das in allen Texten vorkommt, den Wert null erhalten und gegebenenfalls im nächsten Schritt entfernt werden. Für weitere Möglichkeiten zur Gewichtung wird auf (Manning & Schutze 1999, 543) verwiesen.

Die Darstellung eines Korpus' als gewichtete TDM hat zwei Nachteile: Je nach Größe des

Korpus' und der Variabilität des Vokabulars[85] kann eine TDM sehr groß werden und damit gegebenenfalls die Hardware ausreizen und die Analysezeiten von HanConc erhöhen. Ebenso werden die den Texten zugrundeliegenden Konzepte und semantische Verbindungen zwischen einzelnen Wörtern unzureichend herausgearbeitet. Beide Herausforderungen können durch eine Singular Value Decomposition (SVD) gelöst werden, deren Zielsetzung einem Clusteringverfahren ähnelt. Während etwa bei einer Principal Component Analysis (PCA) nur korrelierende Spalten zusammengefasst werden (Deerwester et al. 1990, Manning & Schutze 1999), sollen bei einer SVD latente Konzepte entlang beider TDM Achsen aufgedeckt werden.

Eine SVD reduziert die Wörter (w) und die Texte (t) auf k Dimensionen zu den orthonormalen Matrizen U und V. k entspricht der Anzahl an Themen des Korpus, wobei zu bedenken ist, dass k entlang beider TDM Achsen verstanden werden muss. Es werden also nicht nur die unterschiedlichen latenten Themen der einzelnen Texte abgebildet, sondern ebenso die zugehörigen Wortfelder. Soweit nicht anders vorgegeben, berechnet das LSA Paket k als Minimum aus t und w. Die absteigend sortierte Diagonalmatrix wird durch Σ repräsentiert. Durch die Multiplikation von U, Σ und V, wobei V transponiert wird, entsteht eine neue Matrix mit den Dimensionen der ursprünglichen TDM (Deerwester et al. 1990, Manning & Schutze 1999, Venables & Ripley 2002, Katz & Giesbrecht 2006):

$$M_{txw} = U_{txk}\Sigma_{kxk}(V_{wxk})^T \tag{5.15}$$

Die Funktionsweise einer SVD soll an einem simplen Beispiel demonstriert werden: Tabelle 5.9 zeigt einen zufälligen Datensatz für eine TDM mit neun Texten und neun Wörtern. Die einzelnen Zellen repräsentieren die gewichteten Frequenzen pro jeweiligem Text und Wort. Der Datensatz zeigt deutlich zwei thematische Schwerpunkte. Die Worte eins bis fünf kommen nur in den Texten eins bis fünf vor. Die Worte sechs bis neun gibt es ebenso nur in den Texten sechs bis neun. Es kann somit von einer klaren thematischen Trennung ausgegangen werden. Allerdings enthält Text sechs die Wörter vier und fünf. Es ist somit nicht eindeutig, zu welchem Themenblock dieser Text gehört.

Tabelle 5.9: Beispieldatensatz einer TDM zur Demonstration einer SVD

	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6	Text 7	Text 8	Text 9
Wort 1	0,830	0,730	0,980	0,470	0,810	0	0	0	0
Wort 2	0,210	0,510	0,930	0,830	0,710	0	0	0	0
Wort 3	0,280	0,890	0,760	0,160	0,730	0	0	0	0
Wort 4	0,250	0,510	0,620	0,610	0,250	0,210	0	0	0
Wort 5	0,080	0,810	0,040	0,540	0,270	0,350	0	0	0
Wort 6	0	0	0	0	0	0,810	0,690	0,890	0,040
Wort 7	0	0	0	0	0	0,130	0,070	0,040	0,620
Wort 8	0	0	0	0	0	0,100	0,380	0,840	0,830
Wort 9	0	0	0	0	0	0,040	0,930	0,050	0,190

[85] Unzureichende Textaufbereitungsschritte führen ebenfalls zu diesem Effekt.

Tabelle 5.10 zeigt eine Korrelationsmatrix auf Spalten-/Textebene. Die Texte eins bis fünf und sechs bis neun des obigen Beispiels haben jeweils einen hohen positiven Korrelationskoeffizienten zueinander, wobei ihre Themenblöcke zueinander hoch negativ korrelieren. Text sechs jedoch zeigt Abweichungen. Er verhält sich mit negativen Korrelationskoeffizienten zu den ersten fünf Texten so, wie es zu erwarten war. Gegenüber Text sieben und acht ist der Korrelationskoeffizient erwartungsgemäß positiv. Allerdings zeigt sich gegenüber Text neun eine Abweichung. An dieser Stelle ist der Koeffizient negativ.

Tabelle 5.10: Korrelationsmatrix zu Tabelle 5.9

	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6	Text 7	Text 8	Text 9
Text 1	1	0,630	0,810	0,470	0,800	−0,390	−0,500	−0,410	−0,450
Text 2		1	0,690	0,660	0,830	−0,300	−0,730	−0,600	−0,660
Text 3			1	0,690	0,930	−0,500	−0,610	−0,510	−0,560
Text 4				1	0,640	−0,230	−0,650	−0,540	−0,590
Text 5					1	−0,490	−0,640	−0,530	−0,590
Text 6						1	0,360	0,610	−0,150
Text 7							1	0,530	0,200
Text 8								1	0,450
Text 9									1

Es soll nun überprüft werden, ob eine SVD die Trennschärfe der Themenblöcke erhöht. Entsprechend der Vorgehensweise aus Deerwester et al. (1990) wird die ursprüngliche Matrix umgewandelt zu:

Tabelle 5.11: U zu Tabelle 5.9

−0,580	0,040
−0,500	0,030
−0,460	0,030
−0,350	−0,030
−0,280	−0,070
−0,020	−0,690
0	−0,190
−0,010	−0,600
0	−0,350

Tabelle 5.12: Diagonalmatrix aus Σ zu Tabelle 5.9

2,930	0
0	1,840

Tabelle 5.13: V transponiert zu Tabelle 5.9

−0,280	−0,510	−0,550	−0,390	−0,450	−0,070	−0,010	−0,010	0
0,020	0	0,040	0	0,030	−0,370	−0,570	−0,620	−0,390

Werden diese drei Matrizen miteinander multipliziert, ergibt sich eine Matrix mit den Dimensionen der Ursprungsmatrix:

Tabelle 5.14: SVD Matrix der Beispielmatrix zu Tabelle 5.9

	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6	Text 7	Text 8	Text 9
Wort 1	0,480	0,860	0,940	0,650	0,770	0,080	−0,030	−0,030	−0,020
Wort 2	0,420	0,750	0,810	0,570	0,670	0,080	−0,020	−0,020	−0,020
Wort 3	0,380	0,680	0,740	0,520	0,610	0,070	−0,020	−0,020	−0,020
Wort 4	0,290	0,530	0,570	0,400	0,470	0,080	0,030	0,040	0,020
Wort 5	0,230	0,410	0,440	0,310	0,360	0,100	0,080	0,080	0,050
Wort 6	0	0,030	−0,010	0,030	0	0,480	0,720	0,780	0,490
Wort 7	0	0,010	−0,010	0,010	0	0,130	0,200	0,220	0,140
Wort 8	−0,010	0,010	−0,030	0,010	−0,020	0,410	0,620	0,680	0,420
Wort 9	−0,010	0,010	−0,020	0,010	−0,010	0,240	0,370	0,400	0,250

Im Vergleich zur Urspungsmatrix fallen einige Veränderungen auf. So wurden die Wertspitzen beispielsweise abgemildert. Vor allem sehr hohe Werte wie Text 3/Wort 1 mit einem Wert von 0,98 oder Text 3/Wort 2 mit einem Wert von 0,93 wurden auf 0,94 und 0,81 abgemildert. Gleichzeitig ist in den Bereichen, die nicht von den Themenblöcken besetzt sind und damit ursprünglich den Wert null hatten, ein Rauschen im Bereich von -0,03 bis 0,08 entstanden. Die Worte vier und fünf in Text sechs, die als Testrauschen hinzugefügt wurden, sind zu dem oben benannten Rauschen herabgestuft worden.

Tabelle 5.15 zeigt die Korrelationsmatrix der SVD Matrix. Es ist zu beachten, dass die einzelnen Werte auf die zweite Nachkommastelle gerundet sind. Im Vergleich zu Tabelle 5.10 fallen die erhöhten Korrelationskoeffizienten auf. Während in Tabelle 5.10 die Koeffizienten für Thema eins bei 0,63 und 0,93 liegen, korrelieren die selben Texte nach der SVD nahezu perfekt. Gleiches gilt für Thema zwei. Auch hier wurden teils negative Koeffizienten zu nahezu perfekt positiven Koeffizienten umgewandelt. Die Abgrenzung zwischen den einzelnen Themen ist ebenso klarer. Die Koeffizienten sind hier von -0,39 bis -0,73 auf jeweils mindestens -0,74 gesunken. Somit hat die SVD als Datenaufbereitungsschritt die Qualität der Daten merklich erhöht. Da es sich bei einer SVD um ein unüberwachtes Lernverfahren handelt, kann die Themenanzahl bei einem realen Datensatz nicht a-priori bestimmt werden.

Tabelle 5.15: Korrelationsmatrix zu Tabelle 5.14

	Text 1	Text 2	Text 3	Text 4	Text 5	Text 6	Text 7	Text 8	Text 9
Text 1	1	1	1	1	1	−0,750	−0,840	−0,840	−0,850
Text 2		1	1	1	1	−0,740	−0,830	−0,830	−0,840
Text 3			1	1	1	−0,750	−0,840	−0,840	−0,850
Text 4				1	1	−0,730	−0,830	−0,830	−0,840
Text 5					1	−0,750	−0,840	−0,840	−0,850
Text 6						1	0,990	0,990	0,990
Text 7							1	1	1
Text 8								1	1
Text 9									1

Der Nutzen einer so bearbeiteten TDM ergibt sich aus dem Vergleich einzelner Vektoren. Je nach Anwendungsfall werden entweder Spalten oder Zeilen miteinander verglichen. Auf Spaltenebene ergibt sich aus dem Vergleich die thematische Ähnlichkeit zweier Texte. Die obigen Tabellen und Erklärungen verdeutlichen diese Herangehensweise. Werden die Zeilen miteinander verglichen, ergibt sich daraus die semantische Ähnlichkeit der Wörter. Beide Vergleiche können mit unterschiedlichen Verfahren angestellt werden. Allen Verfahren ist dabei gemein, dass sie Zeilen und Spalten jeweils als Vektoren betrachten. Das LSA Paket in R verwendet als Vergleichsverfahren den Kosinus mit folgender Formel:

$$Kosinus(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.16}$$

Der Kosinus ist zwischen null und eins definiert. Damit kann er ähnlich eines Korrelationskoeffizienten gelesen werden. Höhere Werte bedeuten, dass die zwei übergebenen Vektoren, d.h. Texte oder Wörter, eine entsprechend hohe Ähnlichkeit haben. HanConc benutzt hierfür die $associate$ Funktion des LSA R Pakets. Dieses berechnet eine komplette Korrelationsmatrix und gibt die Werte zum Suchbegriff in absteigender Reihenfolge zurück. Die obersten fünf Werte werden an das Frontend übergeben.

Um die Laufzeit zu reduzieren, wurden alle Schritte bis zur fertigen TDM vorberechnet. Beim Starten von HanConc werden diese TDM in den RAM geladen und zur Laufzeit wird nur die oben erwähnte $associate$ Funktion aufgerufen.

4. Darstellung im Frontend

Die Darstellung der Ergebnisse aus der LSA im Frontend ist stark reduziert. Die oben beschriebene Funktion gibt eine Liste mit fünf Wörtern und den dazugehörenden Kosinuswerten zurück. Diese Werte werden mit 100 multipliziert und als Prozentwert neben den jeweiligen Wörtern dargestellt.

5. Nutzen in der Schreibberatung

HanConc bietet, wie bereits oben beschrieben, keine „Warenkorbanalysen" an. Unabhängig von technischen Limitierungen führt vor allem das dezentrale Design der Anwendung dazu, dass nur unzureichend Benutzer_innendaten gesammelt werden können und somit keine Datengrundlage für eine solche Analyse besteht. Die LSA stellt an dieser Stelle eine Alternative dar. Anstatt entsprechend eines eCommerce-Systems zu formulieren „andere Kund_innen kauften auch", wird linguistisch „andere Schreibende benutzten in ähnlichen Kontexten folgende Wörter" formuliert.

Pädagogische bzw. Fachliteratur zu Schreibberatung, die sich mit LSA beschäftigt, tut dies vor allem aus dem Blickwinkel einer Synonymanalyse (Turney 2001, Steinhart 2001) oder zur Textbewertung (Burstein & Chodorow 2010, Miller 2003). Diese Herangehensweise ist im Zusammenhang mit dem in Kapitel 2 beschriebenen Vorgehen nicht sinnvoll. Eine LSA findet Wörter, die in ähnlichen Kontexten eingesetzt werden. Synonyme gehen jedoch darüber hinaus und sind als referenziell gleichwertig definiert. Am Beispiel von „Versuch", „Experiment" und

„Test" soll verdeutlicht werden, dass dies bei wissenschaftlichen Texten jedoch nicht der Fall ist. Der Duden gibt alle drei Wörter als Synonyme an (Dudenredaktion 2006). Würde dies für alle wissenschaftlichen Texte gleichermaßen zutreffen, müsste von einer Gleichverteilung innerhalb aller Subkorpora und über alle Subkorpora hinweg ausgegangen werden und in der LSA würden die einzelnen Wörter gegenseitig als ähnlich erkannt werden. In diesem Fall soll sich die Untersuchung auf das Wort „Versuch" konzentrieren und die drei ingenieurwissenschaftlichen Korpora betrachten. Beide Vorgehensweisen, sowohl intra- als auch interkorpusspezifisch, können zur Beantwortung der Synonymfrage herangezogen werden. Der interkorpusspezifische Ansatz zeigt zusätzlich jedoch auch auf, dass es sich bei den einzelnen Korpora nicht um beliebige Textsammlungen handelt.

Tabelle 5.16 gibt die absolute und relative Frequenz von „Versuch", „Experiment" und „Test" je ingenieurwissenschaftlichem Subkorpus an[86]. Sollte die Festlegung des Dudens auf Synonyme zutreffen, wäre davon auszugehen, dass das Verhältnis der drei Wörter zueinander in etwa gleich bleibt und über die unterschiedlichen Subkorpora hinweg stabil ist.

Tabelle 5.16: Synonyme nach Subkorpus als absolute und relative Frequenz (in ‰)

Wort	FBau		FElt		FMas	
	Anzahl	Rel.Freq.	Anzahl	Rel.Freq.	Anzahl	Rel.Freq
versuch	1	0,0002	0	0	0	0
Versuch	596	0,148	100	0,047	435	0,089
experiment	4	0,001	3	0,001	2	0,0004
Experiment	128	0,032	116	0,055	329	0,067
test	29	0,007	7	0,003	19	0,004
Test	236	0,059	130	0,062	82	0,017
Gesamt	4.021.551		2.112.801		4.878.799	

Abbildung 5.22 zeigt die relativen Frequenzen aus Tabelle 5.16. Die Abbildung veranschaulicht, dass die klein geschriebenen Varianten zwar vorkommen, aber kaum ins Gewicht fallen. Auf einen Signifikanztest auf Gleichverteilung sowohl zwischen den einzelnen Korpora als auch zwischen den einzelnen Wörtern wird auf Grund der geringen Stichprobengröße der aggregierten Daten verzichtet.

Die Unterschiede zwischen den Korpora in den Frequenzen von „Versuch", „Experiment" und „Test" können auf zwei Phänomene hindeuten: Bei den untersuchten Wörtern handelt es sich um tatsächliche Synonyme und verschiedene Traditionen der einzelnen Wissenschaftssprachen führen zu den unterschiedlichen Frequenzen oder in den Frequenzunterschieden manifestieren sich die Unterschiede in den wissenschaftlichen Vorgehensweisen.

Tabelle 5.17 zeigt die Anzahl an Texten für das bauingenieurwissenschaftliche Korpus, die eine spezifische relative Häufigkeit der untersuchten potentiellen Synonyme im Verhältnis zur Satzanzahl aufweist. Die Häufigkeiten wurden zu je 0,1 in Kategorien eingeteilt, wobei nicht jede Kategorie vergeben ist. Jede Zeile in der Kategoriespalte gibt jeweils nur das obere Limit

[86] Der Vollständigkeit halber werden auch die kleingeschriebenen Schreibweisen betrachtet.

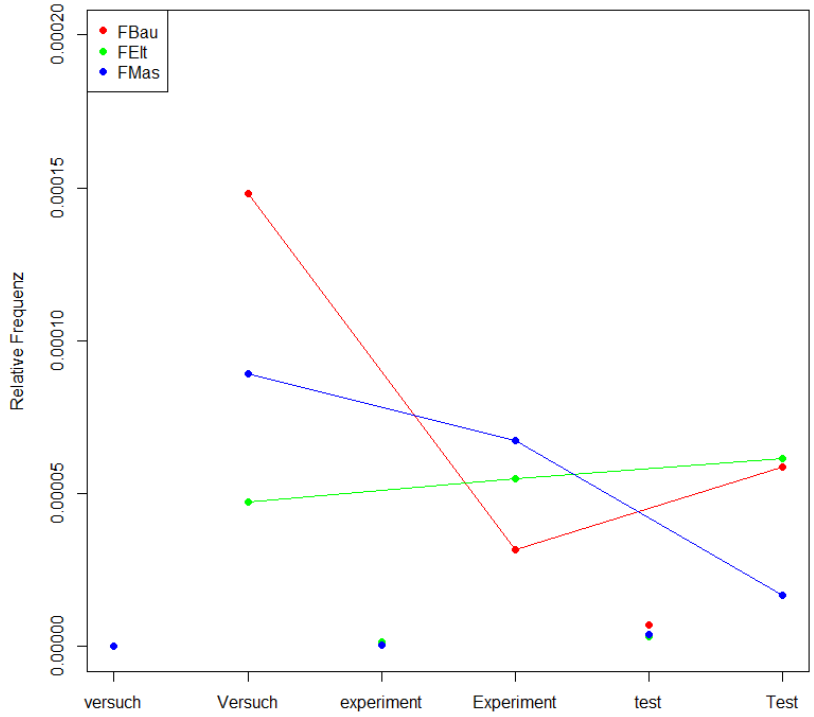


Abbildung 5.22: Grafische Darstellung von Tabelle 5.16

an und das untere Limit ergibt sich aus der darüberliegenden Zeile. Auf Basis dieser Tabelle und der jeweils auf die gleiche Weise erstellten Tabellen für das FElt und FMas Korpus können Signifikanztests in Form von Kolmogorov-Smirnov- und χ^2-Tests durchgeführt werden.

Für das Wort „Versuch" ergibt sich die Testmatrix in Tabelle 5.18. Diese Tabelle stellt die entsprechenden Spalten für das FBau, FElt und FMas Korpus analog zu Tabelle 5.17 gegenüber und bildet die Grundlage für die Signifikanztests, deren Ergebnisse in Tabelle 5.19 inklusive der χ^2-, D- und p-Werte dargestellt werden. Bei beiden Signifikanztests wurden 30 Freiheitsgrade gewählt. Für den Vergleich von „Versuch" im FElt und FMas Korpus wurde die Anzahl an Freiheitsgraden auf 25 reduziert, da es in mehreren Zellen für beide Korpora keine Einträge gibt. Auffällig ist, dass der Kolmogorov-Smirnov Test keine signifikanten Unterschiede feststellen kann, während der χ^2 Test in allen Fällen signifikant ist.

Die Diskrepanz zwischen den Signifikanztests lässt sich aus ihrer Berechnungsmethode erklären: Der Kolmorogov-Smirnov Test untersucht die maximale Distanz der beiden kumulierten Dichtefunktionen. Da die Unterschiede jenseits der ersten beiden Kategorien nur marginal sind, ist der Test insgesamt auch nicht signifikant. Der χ^2 Test hingegen berechnet den erwarteten Wert für jede Zelle unter der Annahme, dass beide beobachteten Verteilungen der gleichen theoretischen Verteilung folgen. Die summierte Differenz aus erwarteten und tatsächlichen Werten ergibt den χ^2 Wert. Durch dieses Vorgehen wird die gesamte Verteilung und nicht nur die maximale Differenz berücksichtigt (Sheskin 2003).

Zur Verdeutlichung zeigt Abbildung 5.23 die kumulierten Dichtefunktionen zu Tabelle 5.18.

Tabelle 5.17: Anzahl an Texten im FBau Korpus mit spezifischem Verhältnis vom potentiellen Synonym zur Satzanzahl als relative Frequenz

Kategorie	versuch	Versuch	experiment	Experiment	test	Test
0	94	39	92	65	83	51
$\leq 0,01$	1	32	3	19	9	29
$\leq 0,02$	0	8	0	3	1	4
$\leq 0,03$	0	2	0	0	0	3
$\leq 0,04$	0	6	0	1	0	3
$\leq 0,05$	0	0	0	0	1	1
$\leq 0,06$	0	0	0	0	0	2
$\leq 0,07$	0	0	0	1	0	0
$\leq 0,08$	0	0	0	2	0	1
$\leq 0,10$	0	1	0	1	0	0
$\leq 0,12$	0	1	0	0	0	0
$\leq 0,14$	0	0	0	1	0	0
$\leq 0,15$	0	1	0	0	0	0
$\leq 0,16$	0	1	0	0	0	0
$\leq 0,18$	0	1	0	0	0	0
$\leq 0,20$	0	1	0	0	0	0
$> 0,2$	0	2	0	2	1	1

Tabelle 5.18: Anzahl an Texten im FBau, FElt und FMas Korpus mit spezifischem Verhältnis von der Häufigkeit des Vorkommens von „Versuch” zur Satzanzahl als relative Frequenz

Kategorie	Versuch (FBau)	Versuch (FElt)	Versuch (FMas)
0	39	30	52
$\leq 0,01$	32	9	60
$\leq 0,02$	8	3	17
$\leq 0,03$	2	3	1
$\leq 0,04$	6	0	8
$\leq 0,05$	0	1	0
$\leq 0,06$	0	0	2
$\leq 0,07$	0	0	1
$\leq 0,08$	0	0	1
$\leq 0,09$	0	0	0
$\leq 0,10$	1	1	0
$\leq 0,12$	1	0	0
$\leq 0,13$	0	0	0
$\leq 0,14$	0	0	0
$\leq 0,15$	1	0	0
$\leq 0,16$	1	0	0
$\leq 0,18$	1	0	0
$\leq 0,19$	0	0	1
$\leq 0,20$	1	0	0
$> 0,20$	2	2	0

Tabelle 5.19: Ergebnisse eines χ^2 und Komogorov-Smirnov Tests zur quantitativen Untersuchung der Verwendung des Worte „Versuch" im FBau, FElt und FMas Korpus

	$\chi 2$	p-Wert	D	p-Wert
FBau zu FElt	65,29	$2{,}01 \cdot 10^{-4}$	0,25	0,56
FBau zu FMas	86,08	$5{,}51 \cdot 10^{-6}$	0,15	0,98
FElt zu FMas	53,44	$5{,}29 \cdot 10^{-3}$	0,1	1

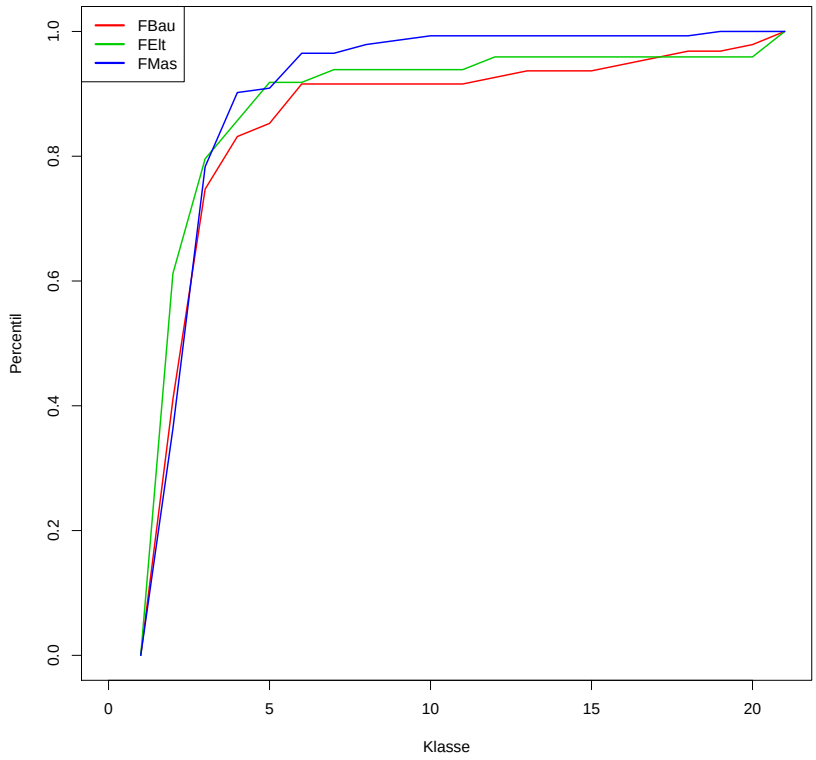


Abbildung 5.23: Kumulierte Dichtefunktion zur quantitativen Untersuchung der Verwendung des Wortes „Versuch" im FBau, FElt und FMas Korpus

Der relativ große Abstand am Anfang der Funktion ist nicht ausreichend, um mit dem Kolmogorov-Smirnov Test einen signifikanten Unterschied zu finden. Da für viele Kategorien keine Werte vorliegen, bleiben die Unterschiede zwischen den einzelnen Verteilungen im Bereich der höheren Kategorien relativ konstant. Dies führt dazu, dass die Unterschiede über einen längeren Zeitraum bestehen und somit kontinuierlich den χ^2 Wert erhöhen.

Die zuvor beschriebenen Untersuchungen, basierend auf relativen Häufigkeiten und Signifikanztests, führen nicht zu eindeutigen Ergebnissen. In einem nächsten Schritt sollen daher Kollokationen als weitere Alternative betrachtet werden (siehe auch Kapitel 5.3.8.3).

Die Tabellen 5.20 und 5.21 zeigen, welche Kollokationen in den drei ingenieurwissenschaftlichen Korpora eine bzw. zwei Positionen links von „Versuch" auftreten. Die Kollokationen werden pro Korpus ausgezählt, absteigend sortiert und auf die häufigsten zehn Kollokationen limitiert dargestellt. Unabhängig von der Position bestehen die häufigsten Kollokationen nur

aus Präpositionen, Artikeln und Auxiliarverben. Ausgenommen davon sind die Kollokationen „Zulaufkanal" (zweite Position links im FBau Korpus) und „Versuch" (zweite Position links im FElt Korpus).

Tabelle 5.20: Kollokationen zweite Position links von „Versuch" nach Korpus

Wort (FBau)	Anzahl (FBau)	Wort (FElt)	Anzahl (FElt)	Wort (FMas)	Anzahl (FMas)
im	28	den	7	bei	35
bei	19	bei	5	Bei	20
in	19	die	5	nach	20
nach	19	LEP	5	im	16
dem	15	Bei	4	fuer	15
wurde	15	Der	3	wurde	15
fuer	14	fuer	3	die	8
aus	12	im	3	Versuch	8
Zulaufkanal	12	in	3	dem	7
In	9	und	3	den	7

Tabelle 5.21: Kollokationen eine Position links von „Versuch" nach Korpus

Wort (FBau)	Anzahl (FBau)	Wort (FElt)	Anzahl (FElt)	Wort (FMas)	Anzahl (FMas)
bei	67	im	11	dem	64
dem	55	der	9	im	39
im	46	und	8	der	37
diesem	44	dem	7	diesem	29
der	39	diesem	7	Der	22
aus	36	den	4	von	14
Der	25	Der	4	Im	8
den	22	diesen	3	nach	8
ersten	17	dieser	3	und	8
in	16	Ein	3	diesen	7

Die Ergebnisse für „Versuch" und „Experiment" unterscheiden sich nur in Reihenfolge und Quantität der Kollokationen[87]. Da es sich bei den Kollokationen um grammatikalische Wörter handelt, trägt die Kollokationsanalyse wenig dazu bei, zu entscheiden, ob es einen semantischen Unterschied zwischen „Versuch", „Experiment" und „Test" gibt.

Sowohl die Auswertung mittels deskriptiver Statistik und Signifikanztests als auch mittels Kollokationen und N-Grams hat nicht zu eindeutigen Ergebnissen geführt. Um den Nutzen einer LSA zu demonstrieren, wird überprüft, ob es zwischen „Versuch" und „Experiment" einen semantischen Unterschied gibt und ob die beiden Wörter darüber hinaus in den unterschiedlichen Korpora abweichend verwendet werden.

HanConc bietet in seinem Frontend die Möglichkeit, eine LSA für ein Wort durchzuführen. Für die oben beschriebene Fragestellung muss die Suche also für Kombinationen aus „Versuch"

[87] Gleiches gilt für die N-Grams.

Tabelle 5.22: Ergebnisse der LSA für das Wort „Versuch” im FBau, FElt und FMas Korpus

FBau	Kosinus (in %)	FElt	Kosinus (in %)	FMas	Kosinus (in %)
versuchsdurchfuehrung	98,77%	uer	97,74%	abbauen	85,66%
vertraeglichkeit	98,72%	zusammenhang	97,59%	temperaturabfall	82,73%
temperaturen	98,52%	auftreten	96,87%	partikelform	82,14%
festkoerper	98,32%	sinusfoermigen	96,65%	nmax	82,11%
einwirkung	98,25%	richtung	96,59%	lauf	81,89%
fehlstellen	98,18%	einher	96,47%	tausendstel	81,02%
grundmaterial	98,14%	linear	96,46%	are	80,69%
coating	98,13%	abhaengigkeit	96,37%	ventilbewegung	80,66%
funktionswert	98,10%	weicht	96,15%	durchbrechen	80,19%
widerstandes	98,06%	einflusses	96,14%	viertakt	79,83%

Tabelle 5.23: Zusammenfassender Vergleich der Ergebnisse einer LSA für das Wort „Versuch" zwischen dem FBau, FElt und FMas Korpus

Fakultät	Ja	Nein	%	Durchschnittlicher Kosinus der gemeinsamen assoziierten Wörter in %
FBau vs. FElt	1.305	2.146	60,81%	75,88%
FElt vs. FMas	685	38.791	1,76%	69,39%
FBau vs. FMas	130	3.321	3,91%	75,42%

und „Experiment" und den einzelnen Korpora durchgeführt werden. Zusätzlich wird noch eine Auswertung auf Basis der gesamten Daten mittels der R Kommandozeile ohne HanConcs Frontend durchgeführt. Tabelle 5.22 zeigt die Ergebnisse einer LSA für das Wort „Versuch" in allen drei ingenieurwissenschaftlichen Korpora. Es ist auffällig, dass die assoziierten Wörter etwa dem erwarteten Vokabular des verwendeten FBau Korpus entsprechen: „Temperaturen" zeigen eine „Einwirkung" auf den „Widerstand" eines „Festkörpers". Ob sich die Kontexte tatsächlich ähneln oder nur durch Zufall fünf Wörter angezeigt werden, die den Erwartungen an die Kontexte in den einzelnen Fakultäten entsprechen, wird mit einer tiefergehenden Analyse untersucht.

Tabelle 5.22 zeigt nur einen Ausschnitt der gesamten LSA. Tabelle 5.23 verdeutlicht hingegen, wie groß die Überschneidungen der zu „Versuch" assoziierten Wörter im Vergleich zwischen den einzelnen Fakultäten ist. 1.305 oder 64,67% der Wörter, die im FBau Korpus mit „Versuch" assoziiert sind, haben auch im FElt Korpus hohe Kosinuswerte. Die übrigen Vergleiche der Fakultäten zeigen jedoch mit 1,76% und 3,91% kaum Übereinstimmungen. Daher liegt die Vermutung nahe, dass das Wort „Versuch" an der FBau und FElt in ähnlichen Kontexten verwendet wird und an der FMas eher nicht.

Sollte es sich bei dem Wort „Experiment" um ein Synonym von „Versuch" handeln und beide Wörter semantisch gleichwertig ausgetauscht werden können, so müsste die LSA ähnliche Kontexte finden. Tabelle 5.24 zeigt allerdings, dass die kontextuellen Überschneidungen von „Versuch" und „Experiment" in den drei Korpora geringer und die Bindungen im Durchschnitt auch schwächer sind. Aus den Tabellen 5.23 und 5.24 ist zu entnehmen, dass es für jedwede Kombination deutlich mehr unterschiedliche als gemeinsame Kontexte gibt. Es ist daher davon auszugehen, dass es sich bei „Versuch" und „Experiment" in den gewählten Korpora nicht um Synonyme handelt und sie außerdem an den verschiedenen Fakultäten in unterschiedlichen Kontexten verwendet werden. Die These, dass es sich bei „Versuch" und „Experiment" um Synonyme handelt, kann zumindest in diesen fachwissenschaftlichen Kontexten empirisch nicht nachgewiesen werden.

Insgesamt sollte eine LSA nicht als finale Antwort gesehen werden. Im Gegensatz zu statistischen Verfahren wie χ^2-Tests geben die Ergebnisse Raum für Interpretationen. Eine LSA begründet einen semantischen Raum für untersuchte Wörter und da auch die anderen Worte in diesem Raum einer Interpretation bedürfen, sollte eine LSA als Startpunkt für weitere Analysen genutzt werden.

Am Beispiel von „Versuch" und „Experiment" konnte gezeigt werden, dass eine LSA eine

Tabelle 5.24: Zusammenfassender Vergleich der Ergebnisse einer LSA von „Versuch" und „Experiment" im FBau, FElt und FMas Korpus

Fakultät	Ja	Nein	%	Durchschnittlicher Kosinus der gemeinsamen assoziierten Wörter in %
FBau	588	2.863	20,53%	64,67%
FElt	5.706	33.770	16,80%	64,46%
FMas	24	3.069	0,78%	54,38%

sinnvolle Ergänzung der bisher vorgestellten Ergebnistypen ist. Vor allem in der Kombination kann eine LSA Schreibberater_innen in die Lage versetzen, den erweiterten Kontext von Suchbegriffen zu erfassen, ohne selbst hunderte Zeilen KWIC lesen zu müssen. Außerdem bieten die aufbereiteten TDMs fortgeschrittenen Nutzer_innen viele Möglichkeiten für weitere Analysen.

5.3.8.6 Deskriptive Statistiken

0. Einleitung

Die Grundidee quantitativer Forschung ist es, zu überprüfen, ob eine Maßnahme Einfluss auf ein bestimmtes Phänomen hat. Hierzu wird dieses Phänomen in einer Beobachtungsgruppe und einer Kontrollgruppe gemessen, die Maßnahme auf die Beobachtungsgruppe angewendet und das Phänomen danach erneut gemessen. Hat die Maßnahme einen Einfluss auf das Phänomen, so sollte sich eine Veränderung in den Messwerten zeigen. Bestehen die Gruppen aus Menschen, Tieren oder Pflanzen kann die Umgebung selten so kontrolliert werden, dass der Einfluss einer Maßnahme isoliert werden kann. Statistik hilft an dieser Stelle, zufällige Abweichungen und reale Einflüsse verlässlich zu unterscheiden.

Vor allem in der didaktisch orientierten Linguistik werden statistische Kennzahlen und Tests herangezogen, um einen Vergleich einer nicht muttersprachlichen Gruppe mit einer muttersprachlichen Kontrollgruppe anzustellen (Ishikawa 2009, Granger & Tyson 1996, Müller 2004, Altenberg & Granger 2001, Durrant & Schmitt 2009). Im Falle von HanConc sollen die Statistiken, vor allem in Kombination mit den Plots, dazu genutzt werden, das Suchwort oder den einzelnen Text im Vergleich zum Korpus zu sehen. Die Statistiken sollen eine Indikation geben, wie die Funde auf das Korpus verteilt sind.

1. Input

Die Statistikfunktion in HanConc basiert auf den Ergebnissen der KWIC Suche. Während der Suche wird ein Vektor gefüllt, der für jeden Text zählt, wie häufig das Suchwort beziehungsweise die Suchkombination gefunden wurde. Dieser Vektor wird an die $statisticsFunction$ übergeben, wo er mit der Korpusdatenbank verglichen wird. Die Datenbank enthält unter anderem die Anzahl an Wörtern und Sätzen je Text und kommt auch bei der Bestimmung der Position auf der Wortliste zum Einsatz. Für die absoluten und relativen Frequenzen werden deskriptive Statistiken berechnet und auf diverse Verteilungsfunktionen getestet. Hierzu werden entsprechend der Funktionen die jeweiligen Parameter mittels des $fitdistrplus$ Pakets geschätzt (Delignette-Muller & Dutang 2015) und mit dem Shapiro-Wilk und dem Kolmogorov-Smirnov

Test aus dem $Stats$ Paket überprüft (R Core Team 2021). Die Ergebnisse werden in einen HTML Code eingebettet und an die übergeordnete Funktion zurückgegeben.

2. Linguistik

Bei den Statistiken handelt es sich um mathematische Beschreibungen von textueller Wirklichkeit. Sie dienen vor allem der quantitativen Beschreibung einzelner Texte innerhalb eines Korpus'. Aus linguistischer Perspektive ist vor allem relevant, welches Phänomen statistisch beschrieben wird. In diesem Fall beziehen sich die statistischen Analysen auf die Anzahl an Suchtreffern pro Text im Verhältnis zum jeweiligen Korpus.

3. Statistik

HanConc stellt pro Text mehrere Zahlen zur Verfügung:

- Anzahl an Texten je Korpus
- Anzahl an Sätzen je Text
- Anzahl an Wörtern je Text
- Position des Treffers innerhalb des Textes, d.h. die Satznummer

Auf Basis dieser Zahlen werden erste Statistiken berechnet. Es handelt sich hierbei um grundlegende deskriptive Statistiken, wie sie etwa in statistischer Einführungsliteratur beschrieben werden (Bortz 2010, Gries 2009, Wooldridge 2002, Sheskin 2003, Cameron & Trivedi 2005).

Zusätzlich werden die absoluten und relativen Häufigkeiten auf eine Gleich- und Normalverteilung getestet. Für die Tests auf Gleich-, Poisson- und negative Binomialverteilung wird ein Kolmogorov-Smirnov (Sheskin 2003) und für die Normalverteilung der Shapiro-Wilk Test (Shapiro & Wilk 1965) eingesetzt.

Für die Berechnung des Mittelwertes wird das arithmetische Mittel verwendet, sodass:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{5.17}$$

wobei x_i die $i$te Beobachtung von x und n die Anzahl an x darstellt. Da das arithmetische Mittel durch Extremwerte verfälscht werden kann (Gries 2009, 116), wird zusätzlich noch die Standardabweichung berechnet (Bortz 2010):

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} \tag{5.18}$$

wobei x_i die $i$te Beobachtung von x, n die Anzahl an x und μ das arithmetische Mittel darstellt. Vor allem in Kombination erlauben Mittelwert und Standardabweichung eine Abschätzung der zugrundeliegenden Verteilung. Alternativ könnten auch entsprechende Grafiken geplottet werden. Allerdings wird aus Gründen der Übersichtlichkeit des Frontends darauf verzichtet.

Aus linguistischer Sicht ist vor allem interessant, ob sich die Beobachtungen gleichmäßig über alle Gruppen, d.h. Sätze und Texte, verteilen oder ob es gegebenenfalls Faktoren gibt, welche die Anzahl an Beobachtungen beeinflussen. Eine Gleichverteilung würde bedeuten, dass das gesuchte Wort in allen Texten ungefähr gleich oft vorkommt. Vor allem bei funktionalen

Wörtern wie Artikeln oder Konjunktionen kann von einer Gleichverteilung ausgegangen werden. Beobachtungen für Text a und b sind gleichverteilt, wenn die Wahrscheinlichkeit ihres Auftretens gleich groß ist (Sheskin 2003). Vor allem auf der Contrastive Interlanguage Analysis (CIA) (Granger 1996) basierende Studien gehen von einer nicht gleichverteilten Grundverteilung aus (Ishikawa 2009, Müller 2004, Gärtner 2013, Granger & Tyson 1996).

Neben Gleichverteilungen sind Normalverteilungen Standardverteilungen, die bei vielen natürlichen Phänomenen erwartet werden können (Bortz 2010). So gehen Normalverteilungen davon aus, dass das untersuchte Wort in wenigen Texten sehr häufig, in vielen Texten etwa so häufig wie der Mittelwert und erneut in wenigen Texten eher selten vorkommt. Da es bei Zähldaten nur positive Werte geben kann, funktionieren die Tests auf Normalverteilung nur bei einem ausreichend häufigen Vorkommen des Wortes (Bortz 2010). Die Dichtefunktion der Normalverteilung, wie sie in Gleichung 5.19 dargestellt ist, hängt von zwei Parametern ab: μ wurde bereits in Gleichung 5.17 und σ in Gleichung 5.18 vorgestellt. Mit diesen beiden Parametern und der Dichtefunktion kann eine hypothetische Verteilung erzeugt werden, die mit den tatsächlichen Beobachtungen verglichen werden kann. Ob die Abweichung von der hypothetischen zur tatsächlichen Verteilung signifikant ist, kann mit entsprechenden Tests überprüft werden.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{5.19}$$

Der Shapiro-Wilk Test (Shapiro & Wilk 1965) wurde für HanConc vor allem deshalb ausgewählt, da er auch für Stichproben mit weniger als 50 Texten oder Sätzen angewandt werden kann und dennoch robuste Ergebnisse liefert (Razali, Wah et al. 2011). Die Teststatistik W ergibt sich aus dem Vergleich einer Standardnormalverteilung mit den Beobachtungen. Der Vergleich wird mittels einer Analyse der Varianz (ANOVA) durchgeführt. Grenz- und p-Werte können durch eine Monte-Carlo Simulation auch für größere Werte ermittelt werden (Shapiro & Wilk 1965). W wird wie folgt berechnet:

$$W = \frac{b^2}{\sigma^2} \tag{5.20}$$

wobei σ^2 der Varianz der Funktion entspricht. b^2 wird auf Basis von Zahlenpaaren der sortierten Beobachtungen berechnet. Vom letzten Wert wird der erste Wert abgezogen, vom vorletzten Wert der zweite Wert und so weiter. Die Ergebnisse dieser Subtraktionen werden mit dem entsprechenden Parameter a multipliziert und aufaddiert, sodass:

$$b^2 = \sum_{i=1}^{k} a_{n-i+1}(x_{n-i+1} - x_i) \tag{5.21}$$

Bei ungeraden n wird der Median ignoriert. Die Teststatistik W kann mittels entsprechender Tabellen in einen interpretierbaren p-Wert umgerechnet werden (Shapiro & Wilk 1965).

Zusätzlich zu dem in HanConc verwendeten Shapiro-Wilk Test kann eine Verteilung auch mit einem χ^2, Kolmogorov-Smirnov, Lilliefors, Cramer-von Mises, Anderson-Darling, D'Agost-

ino K^2 und Jarque-Bera Test (Shapiro & Wilk 1965, Razali et al. 2011, D'agostino, Belanger & D'Agostino Jr 1990, Jarque & Bera 1980) auf eine Normalverteilung überprüft werden. Bis auf den D'Agostino K^2, den Jarque-Bera und den Shapiro-Wilk Test können alle übrigen Tests auch auf andere Verteilungen angewendet werden. Kolmogorov-Smirnov und Lilliefors Tests basieren auf der Idee, die hypothetische und die tatsächliche Verteilung übereinanderzulegen und dann den Punkt des größten Abstands zu vergleichen. Ist dieser Abstand groß genug, wird die Nullhypothese auf Normalverteilung abgelehnt. Cramer-von Mises und Anderson-Darling berechnen die Differenz zwischen beiden Verteilungen und bestimmen dann das Integral. Ist dieses größer als der entsprechende Referenzwert, so wird die Nullhypothese abgelehnt. Der D'Agostino K^2 und der Jarque-Bera Test basieren auf den beiden nächst höheren Momenten der Funktion. Wenn Mittelwert und Standardabweichung, bzw. die Varianz, die ersten beiden Momente sind, so handelt es sich bei der Schiefe und der Kurtosis der Funktion um die Momente drei und vier. Die Schiefe einer Verteilung gibt an, ob sie nach rechts oder links geneigt ist, während die Kurtosis beschreibt, wie flach bzw. spitz eine Verteilung ist. Beide Tests überprüfen, ob Schiefe und Kurtosis denen einer Normalverteilung entsprechen (D'agostino et al. 1990, Jarque & Bera 1980).

Aufgrund der einfachen Berechnungsgrundlage, der Robustheit auch bei kleinen Stichproben, der Vorinstallation in R und der effizienten Berechnung zur Laufzeit wurde der Shapiro-Wilk Test für HanConc ausgewählt. Die anderen oben genannten Signifikanztests können an der entsprechenden Stelle im Quellcode als Alternative implementiert werden und den Shapiro-Wilk Test ersetzen.

Im Gegensatz zur Normalverteilung gibt es für die meisten anderen Standardverteilungen keine spezialisierten Tests. Stattdessen werden für diese Verteilungen allgemeine Goodness-of-Fit Tests wie der χ^2 als auch der Kolmogorov-Smirnov Test (Sheskin 2003) eingesetzt. Der χ^2 Test vergleicht dabei die, ähnlich wie bei einem Histogramm, zu Gruppen zusammengefassten Werte mit denen der angenommenen Verteilung. Aus allen Abweichungen wird der χ^2 Wert berechnet, der entsprechend der Anzahl an Gruppen, d.h. Freiheitsgraden, in einen p-Wert umgewandelt werden kann. Der Kolmogorov-Smirnov Test hingegen funktioniert auch mit ordinal und metrisch skalierten Beobachtungen. Hierbei wird die kumulierte Dichtefunktion der beobachteten mit den theoretischen Werten verglichen. Die Teststatistik d ergibt sich aus:

$$d_{max} = |S(X_i - F_0(X_i))|. \tag{5.22}$$

Von jedem tatsächlichen Wert X an der Stelle i wird der hypothetische Wert an ebenjener Stelle abgezogen. Die Differenz wird für die kumulierte Dichtefunktion betrachtet. Der Betrag der Differenz wird genutzt, um sicherzugehen, dass es keinen Einfluss hat, ob die tatsächliche oder die hypothetische Funktion an der Stelle i höher ist. Der maximale Wert über die gesamte Funktion entspricht der Teststatistik d.

Der Kolmogorov-Smirnov Test wird in der aktuellen Version von HanConc für drei Verteilungen genutzt: Die Gleichverteilung geht davon aus, dass jeder Wert gleich wahrscheinlich ist. Sie wurde gewählt, da es sich um eine vergleichsweise einfache Verteilung handelt. Vordring-

liches Ziel ist es, den programmiertechnischen Unterbau für weitere Analysen vorzubereiten. Auf Basis vorheriger Studien des Autors wurden mit der Poisson und der Negativen Binomialverteilung außerdem zwei Zähldatenverteilungen aufgenommen. Es hat sich gezeigt, dass diese beiden Verteilungen vielfach dafür geeignet sind, Zähldaten zu beschreiben und damit die Grundlage für weitere statistische Verfahren wie Generalised Linear Models (GLM) mit Zähldatenkomponenten zu legen (Gärtner 2014, Cameron & Trivedi 2005, Winkelmann 2008). Die Poisson Verteilung ist definiert als:

$$f(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{5.23}$$

wobei $\lambda \epsilon \mathbb{R}$, $x = 0, 1, 2, ...$ und e der Eulerschen Zahl entspricht. Ebenso wie es sich bei μ und σ um Parameter einer Normalverteilung handelt, ist λ ein Parameter für eine Poisson Funktion. Sowohl μ als λ sind die Erwartungswerte ihrer Verteilungen und können auf gleiche Art bestimmt werden. Im Gegensatz zur Normalverteilung entspricht die Varianz einer Poissonverteilung auch dem Erwartungswert. Ist die Varianz größer als der Erwartungswert, kann es sinnvoll sein, eine negative Binomialverteilung zu schätzen (Cameron & Trivedi 2005):

$$f(x) = \binom{r+x-1}{x} p^r (1-p)^x \tag{5.24}$$

Bei r handelt es sich um einen per Maximum Likelihood gefitteten Parameter (Anscombe 1949). Hierfür wird das $fitdistrplus$ Paket in R verwendet (Delignette-Muller & Dutang 2015). p ergibt sich aus μ, indem $p = \frac{r}{\mu + r}$ ist. Unabhängig von der inhaltlichen Interpretation der einzelnen Parameter eignet sich die negative Binomialfunktion vielfach, um Zähldaten mit einem hohen Anteil an Nullwerten zu beschreiben (Gärtner 2014). Auf Basis einer gefitteten Funktion lässt sich in einem weiteren Schritt einfacher abschätzen, welche Art von weiterführender Analyse sinnvoll erscheint. Besonders für die hier beschriebenen Verteilungsfunktionen existieren GLM, die an diese Verteilungen angepasst sind (Cameron & Trivedi 2005, Cameron & Trivedi 2013, Winkelmann 2008, Wooldridge 2002).

Die absoluten Frequenzen des Suchworts pro Text und die relativen Frequenzen in Bezug auf die Wort- und Satzanzahl pro Text werden in HanConc auf Normal- und Gleichverteilung getestet. Da sowohl die Poisson- als auch die negative Binomialverteilung nur für ganzzahlige Werte definiert sind, werden beide Verteilungen nur für die absoluten Frequenzen verwendet. Wird beim Funktionsaufruf der $dunif$, $dpois$ oder $dnbin$ Parameter verändert, kann auf eine der folgenden Verteilungen getestet werden: Beta, Binomial, Cauchy, χ^2, F, Gamma, Geometrisch, Hypergeometrisch, Log-Normal, Multinomial, Negativ Binomial, Normal, Poisson, Students T und Weibull (R Core Team 2021). Es ist dabei zu beachten, dass aus inhaltlicher und statistischer Sicht nicht alle testbaren Verteilungen gleich wahrscheinlich sind.

4. Darstellung im Frontend

Die Darstellung im Frontend erfolgt in tabellarischer Form. Folgendes Beispiel zeigt eine solche Auswertung:

```
Statistics
Descriptive Statistics

Mean per Text (μ):  2.346939
SD per Text (σ):  4.125889

Mean per Words (μ): 6.723018e-05
SD per Words (σ): 0.0001393079

Mean per Sentences (μ): NaN
SD per Sentences (σ): NA

Hits/Words :  5.442999e-05
Hits/Sentences :  Inf

Tests for Empirical Distributions
Absolute Frequencies
Normal Distribution (Shapiro-Wilk Test)
W = 0.6322896 p-Value = 7.822689e-10
Uniform Distribution (Kolmogorov-Smirnov Test)
D = 0.5510204 p-Value = 2.39031e-13
Poisson Distribution (Kolmogorov-Smirnov Test)
D = 0.4553589 p-Value = 2.992018e-09
Negative Binomial Distribution (Kolmogorov-Smirnov Test)
D = 0.4705884 p-Value = 7.512463e-10

Relative Frequency per Words
Normal Distribution (Shapiro-Wilk Test)
W = 0.5512217 p-Value = 5.168004e-11
Uniform Distribution (Kolmogorov-Smirnov Test)
D = 0.9993715 p-Value = 0
Relative Frequency per Sentences
Normal Distribution (Shapiro-Wilk Test)
W = NA  p-Value = NA
Uniform Distribution (Kolmogorov-Smirnov Test)
D = 1 p-Value = 0
```

Quellcode 5.11: Beispielhafte Darstellung der deskriptiven Statistikfunktion von HanConc

Ziel der Statistikfunktion ist es, eine Indikation über die Verteilung der Suchergebnisse zu erlangen, die über einfach Plots hinausgeht. Da die Interpretation der Ergebnisse statistische Kenntnisse verlangt und sie kaum generalisiert werden kann, wurde auf eine Vereinfachung, wie sie bei den Wortassoziationen und Lesarten vorgenommen wird, verzichtet. Dementsprechend werden die Ergebnisse weder in das deutsche Zahlenformat überführt noch gerundet und die Fachbegriffe werden nicht aufbereitet oder vereinfacht.

5. Nutzen in der Schreibberatung

Im Gegensatz zu den vorherigen und den noch folgenden Funktionen von HanConc zielen

die deskriptiven Statistiken bewusst auf fortgeschrittene Nutzer_innen ab. Während die anderen Funktionen die teils komplexen mathematischen und linguistischen Hintergründe gezielt im Frontend vereinfachen, zeigt die Statistikfunktion direkt die Ergebnisse. Die Beweggründe ergeben sich aus der Abhängigkeit der einzelnen Kennzahlen von der zugrundeliegenden Verteilung. Die Kennzahl „Standardabweichung bei absoluter Anzahl" etwa sollte nur vor dem Hintergrund einer Normalverteilung interpretiert werden. Es erscheint daher fahrlässig, diese Kennzahl durch eine Programmroutine interpretieren zu lassen und sie kommentarlos den Nutzer_innen zu präsentieren. Daher wurde entschieden, die Rohdaten anzeigen zu lassen und es den Nutzer_innen zu überlassen, diese zu interpretieren. Wie dies geschehen kann, wird im folgenden Beispiel gezeigt.

Es soll überprüft werden, ob in Arbeiten zum Thema Elektrotechnik eher das Wort „daher" oder „deshalb" verwendet wird. Außerdem soll eingeschätzt werden, ob die sich aus dem Gebrauch dieser Worte ergebenden Konsekutivsätze als Nebensätze formuliert werden oder sich als Hauptsatz auf den vorherigen Satz beziehen. Daher wird auch die Groß- und Kleinschreibung beachtet. Tabelle 5.25 zeigt ausgewählte Ergebnisse der Statistikfunktion für die Worte „daher" und „deshalb" jeweils in der Groß- und Kleinschreibung. Textgrundlage ist das FElt Korpus. Dargestellt sind jeweils die deskriptiven Statistiken und die Tests auf Normal- und Gleichverteilung für die absolute und relative Häufigkeit. Die relative Häufigkeit bezieht sich auf die Anzahl an Wörtern im jeweiligen Text. Auf Basis der reinen Anzahl an Treffern lässt sich schon erkennen, dass in beiden Fällen die eingeleitete Folge im gleichen Satz eher als nachstehender Nebensatz formuliert wird. Bei „deshalb" werden 68% der Sätze mit der Konjunktion begonnen; bei „daher" sind es nur 48%.

An diesem Beispiel zeigt sich die Relevanz der deskriptiven Statistiken. In absoluten Zahlen kommt „daher" deutlich häufiger vor, woraus geschlussfolgert werden könnte, dass es „deshalb" als typischeres Wort vorzuziehen ist. Selbst die gemittelte Anzahl pro Text stützt diese These. Die relativen Zahlen in Bezug auf die Gesamtwortanzahl der Texte zeigen jedoch, dass „deshalb" sowohl am Satzanfang als auch als Einleitung für einen Nebensatz etwa doppelt so häufig vorkommt. Dies hängt damit zusammen, dass „daher" vor allem in längeren Texten genutzt wird. Die Standardabweichungen zeigen außerdem, dass „deshalb" im Sinne der Gleichverteilung deutlich stabiler auftritt als „daher". Eine Erklärung für dieses Phänomen könnte sein, dass „deshalb" als Standardvokabel fungiert und „daher" bei längeren Texten nur als Alternative genutzt wird, um Repetitionen zu vermeiden.

Bei allen Tests auf Normal-, Gleich- und Poissonverteilung muss die Nullhypothese abgelehnt werden. Das bedeutet, dass wahrscheinlich keine der drei Verteilungen vorliegt. Die Verteilung der absoluten Häufigkeiten von „daher" folgt allerdings einer negativen Binomialverteilung. Dabei ist es unerheblich, ob „daher" am Satzanfang steht oder nicht. Insgesamt sind die D Statistiken für die Poisson- und negative Binomialverteilung deutlich geringer als bei der Gleich- und Normalverteilung. Dies deutet darauf hin, dass es für linguistische Forschungsvorhaben sinnvoll sein kann, die angestrebten Regressionen auf Zähldatenregressionen zu erweitern.

Tabelle 5.25: Ausgewählte Statistiken für die Wörter „deshalb" und „daher" im FElt Korpus

	Deshalb	deshalb	Daher	daher
Anzahl	115	169	479	983
μ pro Text	2,346	3,44	9,77	20,06
σ pro Text	4,125	5,319	13,37	21,35
μ pro Worte	$6{,}7230 \cdot 10^{-5}$	$9{,}49301 \cdot 10^{-5}$	$2{,}389 \cdot 10^{-5}$	$4{,}361 \cdot 10^{-5}$
σ pro Worte	$1{,}393 \cdot 10^{-5}$	$1{,}5 \cdot 10^{-5}$	$3{,}730 \cdot 10^{-5}$	$4{,}134 \cdot 10^{-5}$
		Absolute Häufigkeit		
Shapiro-Wilk Test				
W	0,6322	0,6805	0,7258	0,8258
p-Wert	$7{,}8226 \cdot 10^{-10}$	$4{,}7865 \cdot 10^{-9}$	$3{,}1194 \cdot 10^{-8}$	$4{,}3935 \cdot 10^{-6}$
Kolmogorov-Smirnov Test (Uniform)				
D	0,5510	0,6530	0,7755	0,8776
p-Wert	$2{,}390 \cdot 10^{-13}$	0	0	0
Kolmogorov-Smirnov Test (Poisson)				
D	0,4554	0,4096	0,4954	0,4518
p-Wert	$2{,}9920 \cdot 10^{-9}$	$1{,}4424 \cdot 10^{-7}$	$7{,}202 \cdot 10^{-11}$	$4{,}0874 \cdot 10^{-9}$
Kolmogorov-Smirnov Test (Negativ Binomial)				
D	0,5465	0,3634	0,2112	0,1110
p-Wert	$3{,}8947 \cdot 10^{-13}$	$4{,}7911 \cdot 10^{-6}$	0,0253	0,5817
		Relative Häufigkeit		
Shapiro-Wilk Test				
W	0,5512	0,6693	0,6413	0,8851
p-Wert	$5{,}1680 \cdot 10^{-11}$	$3{,}0946 \cdot 10^{-9}$	$1{,}0872 \cdot 10^{-9}$	$1{,}856 \cdot 10^{-5}$
Kolmogorov-Smirnov Test				
D	0,9993	0,9993	0,9982	0,9983
p-Wert	0	0	0	0

Wie das Beispiel gezeigt hat, können absolute Häufigkeiten ohne statistischen Kontext zu Fehlschlüssen führen. Auch wenn diese Funktion über die üblichen Funktionen von Korpussoftware hinausgeht, sollte es doch das Ziel von Schreibberater_innen sein, sich entsprechende Statistikkenntnisse anzueignen, um fundierter beraten zu können. Die Tatsache, dass diese Art des statistischen Vorgehens in der Schreibberatungsliteratur gar nicht thematisiert und auch in der Linguistik außerhalb der Korpus- und Computerlinguistik kaum gelehrt wird, zeigt, dass hier noch Forschungs- und Lehrbedarf besteht.

5.3.8.7 Deskriptive Grafiken und Wortwolken

0. Einleitung

Grafiken dienen dort zur Veranschaulichung, wo Ergebnisse von Analysen nicht durch wenige Kennzahlen oder aggregiert dargestellt werden können. HanConc nutzt Grafiken, um drei Fragestellungen zu beantworten:

1. Wird das Suchwort von einigen Autor_innen übermäßig häufig verwendet oder taucht es in allen Texten gleichermaßen auf?
2. In welchem Teil des Textes wird das Suchwort typischerweise verwendet?
3. Welche Wörter befinden sich in der unmittelbaren Nachbarschaft zum Suchwort und wie stark ist die Verbindung?

Die ersten beiden Fragen werden in HanConc jeweils mit deskriptiven Grafiken (Histogrammen) beantwortet, während für Frage drei Wortwolken zum Einsatz kommen.

1. Input

Die im Folgenden beschriebenen Grafiken stützen sich auf die Ergebnisse der Funktionen aus den vorangegangenen Kapiteln.

Die Frage nach der Verteilung des Suchwortes im Korpus (Frage eins) wird durch Analyse der KWIC Daten beantwortet. Hierzu wird, nach dem Durchsuchen eines Textes, die Anzahl an Sätzen mit dem Suchwort durch die Anzahl an Sätzen des entsprechenden Textes dividiert und in einen Vektor geschrieben. Dieser Vektor wird mit der vorinstallierten Histogrammfunktion von R visualisiert und an das Frontend übergeben.

Bei der Analyse der Position des Suchwortes im Text (Frage zwei) wird ebenso vorgegangen wie bei der Analyse der Verteilung des Suchwortes im Korpus. Der einzige Unterschied besteht darin, dass die Satznummer der Fundstelle durch die Gesamtzahl an Sätzen des jeweiligen Textes dividiert und der Vektor damit gefüllt wird.

Die Wortwolken, welche der Lösungsansatz zur Beantwortung der dritten Frage sind, basieren auf den Kollokationen auf den ersten beiden Positionen rechts und links des Suchwortes. Für jede Position wird jeweils eine konzentrische Wortwolke erstellt, in welcher der stärkste Begriff in der Mitte steht. Die Stärke und damit auch die Größe eines Wortes ergibt sich aus dem entsprechenden MI Score (siehe hierzu Kapitel 5.3.8.3). Für die Darstellung wird das R Paket (Fellows 2018) verwendet.

Werden zwei Korpora oder Wörter gleichzeitig untersucht, erzeugt HanConc für jedes Korpus bzw. Wort eigene Histogramme und Wortwolken. Die Plots sind in eigenen Reitern im Frontend organisiert. Bedingen die Suchparameter, dass einzelne Grafiken nicht benötigt werden, wird stattdessen ein leerer Frame zurückgegeben.

2. Linguistik

Die Histogramme in Referenz zu den Häufigkeiten sind eine visuelle Hilfe, um die Verteilung der Ergebnisse auf die Texte zu veranschaulichen und dabei eine ungleichmäßige Verwendung des Suchwortes aufzuzeigen. Eine entsprechende statistische Überprüfung findet parallel dazu mittels eines Kolmogorov-Smirnov Tests auf Gleichverteilung statt (siehe Kapitel 5.3.8.6).

Die Histogramme in Bezug auf die Position innerhalb der Texte gehen von einem ähnlichen strukturellen Aufbau der den Korpora zugrundeliegenden Doktorarbeiten aus. Erst dadurch können diese Histogramme dafür genutzt werden, die Zugehörigkeit eines Wortes zu einem Abschnitt zu bestimmen. Die Gleichartigkeit des Aufbaus der Doktorarbeiten ergibt sich aus den formalen Vorgaben für das Verfassen einer Abschlussarbeit[88]. Es ist anzunehmen, dass sich durch die universitätsweite vergleichbare Strukturierung von Studien- und Forschungsarbeiten auch Übereinstimmungen bezüglich der Vokabulars an spezifischen Positionen ergeben. Dementsprechend kann anhand der Position des Suchbegriffs im Text abgelesen werden, zu welchem Abschnitt der Arbeit dieses Wort wahrscheinlich gehört.

Wortwolken stellen die bereits in Kapitel 5.3.8.3 beschriebenen Kollokationen grafisch dar. Eine zusätzliche linguistische Aufbereitung findet nicht statt.

3. Statistik

Um die Verteilung der Wörter innerhalb der Texte zu visualisieren, werden klassische Histogramme verwendet. Wie zuvor beschrieben ergibt sich die Position des Suchwortes innerhalb des Textes durch den Index des Satzes, in welchem das Worte gefunden wurde, dividiert durch die Gesamtanzahl der Sätze innerhalb des jeweiligen Textes. Zum besseren Verständnis wird die Position zu einem Prozentwert umgewandelt. Die Anzahl an Klassen, zu denen die Werte aggregiert werden, wird mit der Sturges Formel (1926) berechnet:

$$k = 1 + log_2 n \tag{5.25}$$

Die Histogramme, welche die Häufigkeit der einzelnen Wörter in Bezug auf die Textanzahl zeigen, basieren auf Pivottabellen, die mit Rs *table* Funktion erzeugt und über die *barplot* Funktion in Balkendiagramme überführt werden. Das Vorgehen wird gewählt, um zu verhindern, dass bei kleinen Korpora oder wenigen Klassen, diese zu zu wenigen und damit zu stark vereinfachenden Gruppen zusammengefasst werden.

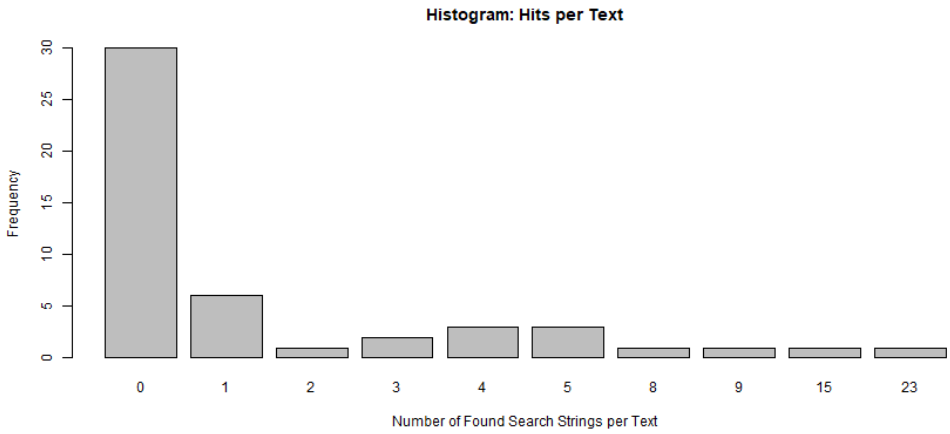
Die Wortwolken basieren auf dem MI Score, welcher in Kapitel 5.3.8.3 erläutert wird.

4. Darstellung im Frontend

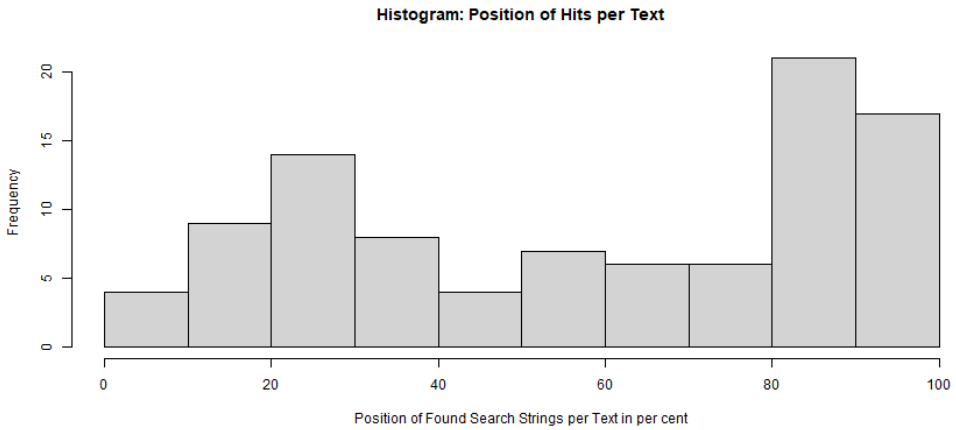
Die Grafiken werden durch HanConcs Frontend in eigenen Reitern dargestellt. Die beiden Histogramme (siehe Abbildung 5.24) befinden sich dabei in einem gemeinsamen und die Wort-

[88] In unterschiedlichem Detailgrad finden sich zahlreiche Vorlagen und Vorgaben zur Struktur der Abschlussarbeiten auf den Internetseiten der einzelnen Institute der Leibniz Universität Hannover (LUH). Trotz der unterschiedlichen wissenschaftlichen Traditionen gibt es einen hohen Grad an Übereinstimmungen.

wolken in einem separaten Reiter (siehe Abbildung 5.25). Um eine möglichst niedrigschwellige Manipulierbarkeit zu gewährleisten, werden Standardfunktionen von R verwendet. Dabei handelt es sich um gerenderte PNG Dateien, die an das Frontend übergeben werden. Aufgrund dieser Struktur sind die Grafiken nicht interaktiv oder manipulierbar. Zur Erhöhung der Einsteigerfreundlichkeit wird auf ästhetisch ansprechendere aber kompliziertere Grafiken auf Basis von JavaScript verzichtet.



(a) Anzahl an Sätzen mit dem Wort „Versuch" im FElt Korpus



(b) Anzahl an Sätzen mit dem Wort „Versuch" an einer bestimmten Position im Text (angegeben als Prozentwert) im FElt Korpus

Abbildung 5.24: Beispiele für deskriptive Grafiken für das Wort „Versuch" im FElt Korpus

Die Darstellung der Wortwolken basiert auf dem WordCloud Paket in R (Fellows 2018). Hierbei ist zu beachten, dass die Wortwolken mittels Position, Größe und Farbe der einzelnen Wörter mehrere Informationen gleichzeitig darstellen können. HanConc unterstützt die Darstellung der 50, im Sinne des MI Score, stärksten Kollokationen. Die Stärke der Kollokation korrespondiert mit der Schriftgröße des Wortes und seiner Position. Die Worte sind konzentrisch und der Stärke nach abnehmend angeordnet. Der Quellcode ist so vorbereitet, dass mittels Übergabe eines weiteren Funktionsparameters die unterschiedlichen Wortarten zusätzlich farbig markiert werden können. Diese Funktion ist jedoch nicht aktiviert, um die Übersichtlichkeit der Darstellung zu gewährleisten.

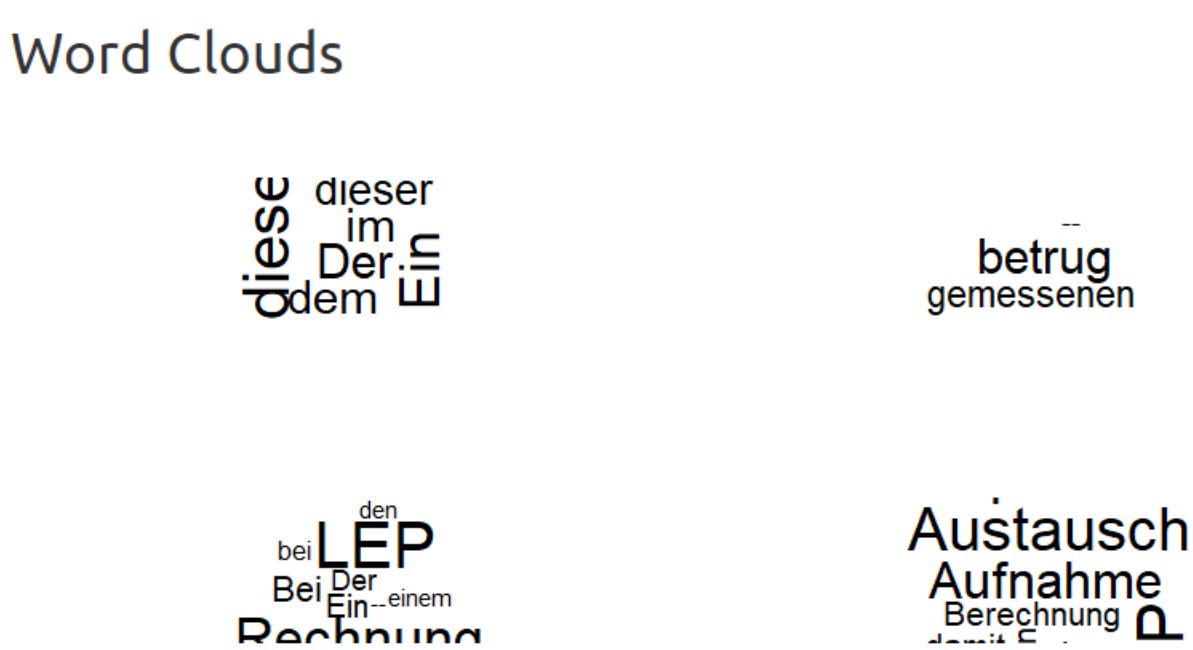


Abbildung 5.25: Wortwolken für das Wort „Versuch” im FElt Korpus

5. Nutzen in der Schreibberatung

Die Grafiken können auf unterschiedliche Weise in der Schreibberatung eingesetzt werden. Während Histogramme eine tiefergehende analytische Funktion haben, die schwer über statistische Kennzahlen dargestellt werden kann, können Wortwolken vor allem zu illustrativen Zwecken eingesetzt werden.

Um die Verteilung eines Wortes auf ein Korpus zu analysieren, können mehrere Kennzahlen zur statistischen Verteilung herangezogen werden. Mittelwert, Standardabweichung und das Ergebnis der Tests auf Gleich- und Normalverteilung (siehe Kapitel 5.3.8.6) beschreiben dies. Allerdings sind diese Zahlen nur im Zusammenspiel verständlich und bedürfen statistischer Vorkenntnisse zur sinnvollen Interpretation. Da HanConc immer auf größtmögliche Einsteigerfreundlichkeit bei gleichzeitiger Möglichkeit zur Komplexität abzielt, sollen die Histogramme eine Interpretation ohne weiteres Vorwissen ermöglichen. Mit ihrer Hilfe soll entschieden werden können, ob es sich bei den Ergebnissen der KWIC um ein statistisches und linguistisches Artefakt handelt oder um eine allgemein verwendete Formulierung. Wird ein Wort oder eine Formulierung nur von einem kleinen Anteil der Schreibenden verwendet, so kann es sich dabei um einen Fachbegriff einer Nischendisziplin handeln oder von persönlichen Vorlieben einzelner Schreibende_r ausgegangen werden (Römer & Arbor 2009). Eine in etwa gleichmäßige Verwendung über das gesamte Korpus hinweg deutet auf eine wissenschaftssprachlich kodifizierte

Formulierung hin (Schroth-Wiechert 2011). Die Unterscheidung zwischen singulärem Fachbegriff, linguistischem Artefakt, überbenutzter Formulierung durch Einzelne und allgemeiner wissenschaftssprachlicher Formulierung obliegt den Studierenden und Schreibberater_innen.

Die zweite Histogramfunktion, die sich mit der Position des Wortes innerhalb der Texte beschäftigt, versucht latente pragmatische Funktionen des Suchbegriffs zu illustrieren. Wie bereits oben beschrieben, wird von einem etwa gleichförmigen Aufbau der Doktorarbeiten ausgegangen. Eine Interpretation kann ähnlich der oben beschriebenen Histogramme erfolgen. Suchbegriffe, die spezielle pragmatische Funktionen innerhalb einer Arbeit einnehmen, etwa zur Beschreibung von Literatur, treten entsprechend gehäuft in vorderen Teilen der Arbeiten auf, während allgemein wissenschaftssprachliche Suchbegriffe gleichverteilt über die gesamte Länge der Arbeiten auftreten.

Wortwolken bieten eine niedrigschwelligere Alternative zu Kollkokationen. Linguistisch und statistisch handelt es sich zwar um die gleichen Daten, jedoch liegt der Unterschied in der Art der Darstellung. Dementsprechend können in Schreibberatungen je nach Vorliebe Kollokationstabellen oder Wortwolken zur Analyse der umstehenden Wörter herangezogen werden.

5.3.8.8 Erweiterter Kontext

Die Funktion „Erweiterter Kontext" stellt eine Erweiterung der KWIC Funktion zur Verfügung, indem nicht nur der Satz mit dem Suchwort, sondern sowohl der vorhergehende als auch der nachfolgende Satz zurückgegeben wird. Trotz Auswahl dieser Funktion bleiben die Ergebnisse der übrigen Funktionen unverändert. Im Zuge der Textaufbereitung kann es dazu kommen, dass Seitenumbrüche, Tabellen, Formeln oder Grafiken Sätze teilen, sodass Satzfragmente als eigenständige Sätze erkannt werden. Die „Erweiterter Kontext" Funktion sollte daher für Situationen genutzt werden, in denen der Satz mit dem Suchwort unvollständig oder uneindeutig ist.

5.3.8.9 Lesarten

0. Einleitung

Ferdinand des Saussures Idee von „signifié" und „signifiant" beschreibt die Beziehung von einem weltlichen Ding zu seiner sprachlichen Referenz. Leider ist diese Beziehung nicht immer eindeutig und manchmal gibt es auch mehrere sprachliche Zeichen, die sich auf das gleiche weltliche Ding beziehen oder mehrere weltliche Dinge, die durch ein sprachliches Zeichen repräsentiert werden. Die Lesartenfunktion in HanConc soll dabei helfen, diese Uneindeutigkeiten durch Analyse des Kontextes auszuleuchten und dadurch aufzeigen, ob ein Begriff sich auf mehrere weltliche Dinge bezieht (De Saussure 2011).

Die in diesem Kapitel vorgestellte Lesartenfunktion sowie die Wortassoziationsanalyse aus Kapitel 5.3.8.5 basieren auf der gleichen linguistischen Idee. In beiden Fällen wird ein Textkorpus als TDM dekonstruiert, mathematisch aufbereitet und dadurch von Störgeräuschen bereinigt. Im Rahmen der Wortassoziationsanalyse wird die Ähnlichkeit einzelner Wörter in Bezug auf ihren Kontext gemessen. Dieser Anwendungsfall, d.h. die Suche nach Synonymen

mittels einer LSA, ist weit verbreitet (Kashyap, Han, Yus, Sleeman, Satyapanich, Gandhi & Finin 2016, Karan, Šnajder & Bašic 2012, Chakrabarti, Chaudhuri, Cheng & Xin 2012, Wang & Hirst 2010, Ekštein & Krčmář 2013). Wird statt der Wortebene die Textebene analysiert, so kann eine LSA im Rahmen des Topic Modellings dazu eingesetzt werden, ein Korpus in Themen zu gruppieren (Hofmann 2000, Bergamaschi & Po 2014, Alghamdi & Alfalqi 2015, Stevens, Kegelmeyer, Andrzejewski & Buttler 2012). Allen Ansätzen ist gemein, dass sie eine Gruppierung der Texte mittels überwachter und unüberwachter Lernverfahren auf Basis einer für eine LSA vorbereiteten TDM vornehmen (siehe Kapitel 5.3.8.5). Die Funktion „Lesarten" gruppiert nun die Texte zu Themenblöcken und kombiniert sie mit der Prototype Theory von Jean Aitchison (2003). In diesem Fall wird also die TDM mittels eines Clusteringverfahrens in Lesarten unterteilt und den Nutzer_innen die Prototypikalität als Prozentwert zurückgegeben. Da alle Sätze das Suchwort enthalten, können die so erzeugten Cluster als Lesarten des selben Wortes betrachtet werden[89].

1. Input

Die Lesartenfunktion basiert auf den KWIC der vorherigen Funktionen. Die einzelnen Sätze werden als DataFrames in einer Liste an die Funktion übergeben. Zusätzlich wird noch die Zugehörigkeit der einzelnen Sätze zu dem jeweiligen Text übermittelt. Auf dieser Basis wird eine TDM konstruiert, welche von deutschen Stopp-Wörtern (etwa Präpositionen, Demonstrativpronomina etc.) und Satzzeichen bereinigt wird. In diesem Fall wird hierfür nicht das *LSA* Paket in R genutzt, sondern die TDM wird selbst konstruiert und anschließend in ein entsprechendes Objekt transformiert, um es mit dem *LSA* Paket weiter zu verarbeiten. Die TDM wird mit dem TF-IDF Algorithmus gewichtet und durch eine SVD geglättet. Um die Themen zu extrahieren, wird ein k-Means Algorithmus zum Clustern und die *calinhara* Funktion des *fpc* Pakets verwendet, um zu überprüfen, ob die Anzahl an Clustern adäquat ist (Hennig 2019). Die Clusterzugehörigkeit wird als zusätzlicher Ergebnistyp an das entsprechende DataFrame angehängt und im Frontend dargestellt.

2. Linguistik

Die in diesem Kapitel und dem Kapitel über Wortassoziationen vorgestellten Techniken dienen der Identifikation von Synonymie und Polysemie bzw. Homonymie. Während die Wortassoziationen dazu genutzt werden, unterschiedliche Wörter in ähnlichen Kontexten zu finden, werden bei der Lesartenfunktion für gleiche Wörter unterschiedliche Kontexte identifiziert. Kontext wird hier als Wortbedeutung verstanden. Im optimalen Fall, das heißt wenn für ein Wort unterschiedliche Sätze mit komplett disjunkten Elementen gefunden werden, kann statistisch von einem Homonym oder Polysem ausgegangen werden. Laut Bußmann & Lauffer (2008) wird zur „Unterscheidung von Polysemie und Homonymie [...] traditionell das Kriterium der etymologischen Verwandtschaft herangezogen" (538). Polysemische Begriffe beruhen demnach auf einer gemeinsamen Wurzel, die sich im Laufe der Zeit ausdifferenziert hat, während homony-

[89] In den oben genannten Quellen zum Thema Topic Modelling werden die Cluster als unterschiedliche Themen innerhalb eines Korpus' interpretiert und zentrale Wörter als Beschreibung der Themen angegeben. Da hier das zentrale Wort, das Suchwort, $a\ priori$ feststeht, werden für die jeweiligen Themenblöcke Beispielsätze zurückgegeben.

me Begriffe, basierend auf unterschiedlichen Herkünften, zufällig gleiche orthografische und phonetische Eigenschaften haben. Da eine etymologische Analyse nicht Teil von HanConc ist, wird an dieser Stelle Homonymie und Polysemie synonym verwendet. Basierend auf der oben skizzierten Definition von Polysem, Homonym und Synonym und der Wortbedeutung-durch-Kontext Definition von Firth (Evert 2005) ergibt sich eine linguistische Indikation der Wortbedeutung. Da die Lesartenfunktion dazu dient, potentielle Bedeutungsunterschiede aufzuzeigen und weniger dazu, diese auszudifferenzieren, erscheint eine oberflächliche Definition von Polysem, Homonym und Synonym entsprechend vergleichbarer Studien wie Sheeba, Vivekanandan, Sabitha & Padmavathi (2013) ausreichend.

3. Statistik

Die Textaufbereitung für die Lesartenfunktion unterscheidet sich, bis auf die oben beschriebenen Ausnahmen, nur unwesentlich von den Aufbereitungen für die Wortassoziationen. In beiden Fällen wird die Textbasis zu einer TDM aufbereitet, diese Mittels TF-IDF gewichtet und mit einer SVD geglättet. Die Unterscheidung beider Funktionen ergibt sich aus der Analyse der aufbereiteten TDM. Während bei den Wortassoziationen einzelne Vektoren auf Wortebene verglichen werden, sollen für die Lesarten Vektoren auf Textebene gruppiert werden, wobei jede dieser Gruppen für eine Lesart des Suchbegriffs steht. Das R Paket *topicmodels* stellt hierfür Algorithmen zur Verfügung (Hornik & Grün 2011). Allerdings ergeben sich zwei Probleme aus der Kombination des *topicmodels* Pakets und HanConc:

Mit der Latent Dirichlet Allocation (LDA) und dem Correlated Topics Model (CTM) werden zwei Algorithmen zur Verfügung gestellt, bei denen es sich um bayes'sche Simulationsmodelle handelt, deren Fitting in *C* implementiert wurde, um eine akzeptable Laufzeit zu ermöglichen. Außerdem wird C++ in der Version 11 verlangt, was abhängig vom Betriebssystem der Anwender_in bzw. des Servers weiteren Aufwand bedeutet.

Zusätzlich zum direkten Nutzen für Schreibberater_innen und ihre Studierenden soll es Interessierten ermöglicht werden, HanConc schnell und zielgerichtet manipulieren zu können. Die Komplexität der Algorithmen und vor allem die Notwendigkeit, passende Hyperparameter voreinzustellen (Hornik & Grün 2011, 8), erschweren daher zusätzlich zu den technischen Hürden den Einsatz des *topicmodels* Pakets.

HanConc setzt daher mit k-Means auf einen einfachen Clustering Algorithmus zur Gruppierung der Wortvektoren, der in R schon vorinstalliert und Teil diverser Online-Tutorien für Datenanalyse und Data Science ist. Der k-Means Algorithmus wird noch zusätzlich mit dem Calinski-Harabasz Index (CHI) kombiniert, der überprüft, ob die Anzahl an Gruppen und damit Lesarten adäquat ist. Wird das Vorgehen des k-Means Algorithmus' mit zwei Dimensionen visualisiert, entspricht er den Darstellungen von Jean Aitchison (2003, 56) in Abbildung 5.26. Jeder Text bzw. Satz wird um einen Prototypen der entsprechenden Lesart angeordnet. Dieser Prototyp steht für das Wesen der Lesart und je weiter ein Satz geografisch von diesem Zentrum entfernt ist, desto unprotypischer ist er (siehe Abbildung 5.26). Die Idee hinter k-Means besteht nun darin, dass es noch k weitere Prototypen geben kann.

Um im Aitchison Beispiel zu bleiben, kann es die Kategorie „Fisch" geben. In mehreren

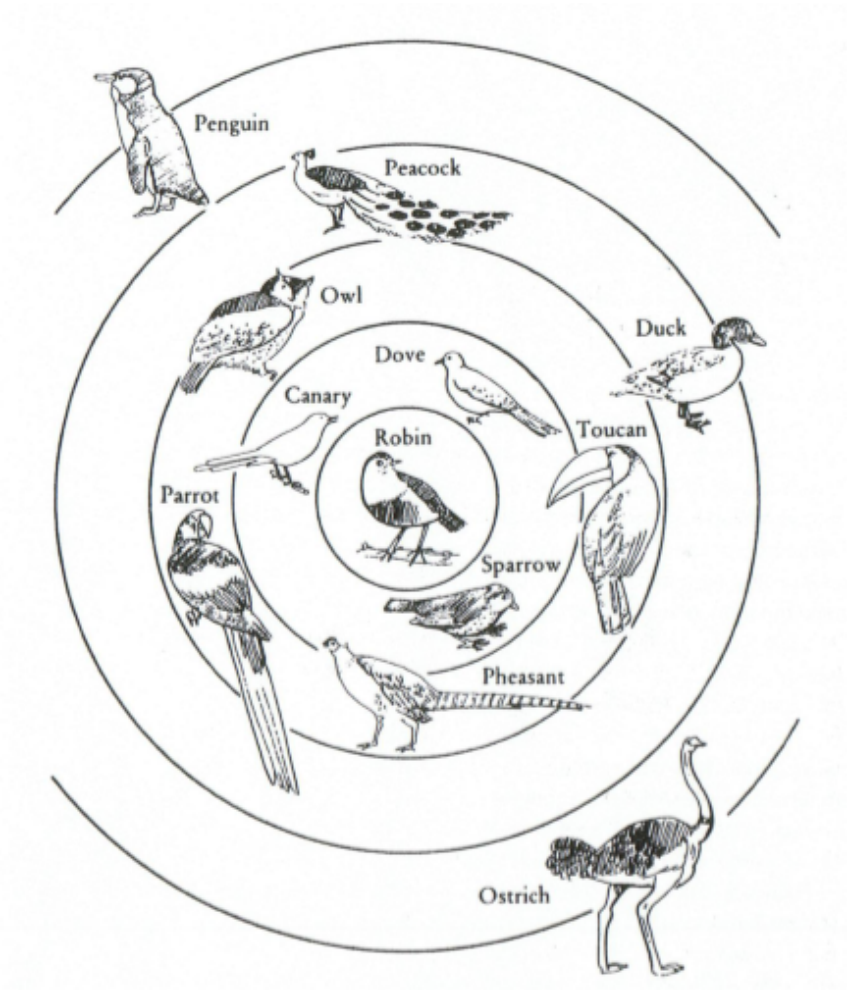


Abbildung 5.26: Birdiness Ranking entnommen aus Aitchison (56, 2003)

Iterationen würde nun ausgehandelt werden, ob ein Pinguin nun eher ein Fisch oder ein Vogel ist. Da er in beiden Fällen vom prototypischen Zentrum entfernt ist, kann erwogen werden, eine dritte Kategorie zu eröffnen. Hieraus ergibt sich das Problem von Clusteringverfahren. Es kann nicht abschließend geklärt werden, wie viele Kategorien sinnvoll sind. Daher muss später bei der Nutzung in der Schreibberatung darauf geachtet werden, dass die Zuordnung einzelner Sätze zu Lesarten als Vorschlag und weniger als Definitivum gedacht ist.

Als k-Means Implementierung wurde die R Übersetzung des Fortran Codes nach Hartigan-Wong gewählt (Hartigan & Wong 1979). Ebenso wie eine lineare Regression basiert k-Means auf der Idee der kleinsten Quadrate. Jeder Punkt M ist Teil eines n-dimensionalen Raumes. In diesem Anwendungsfall befindet sich jeder Satz I in einem n Wörter langen Raum. Ein Cluster L, d.h. eine Lesart, enthält N Sätze. D repräsentiert die euklidische Distanz von einem Punkt I zu einem Cluster L, wobei $I \epsilon M$. Jeder Punkt I wird dem nächstgelegenen Cluster L zugeordnet. Nun werden die Clusterzentren auf Basis der Mittelwerte der zugeordneten Punkte neu berechnet. Jedes Clusterzentrum gehört zu einem lebenden Set. Es soll im nächsten Schritt der minimale quadratische Abstand eines Punktes I zu einem Cluster L gefunden werden:

$$R = \frac{NC_L \cdot D^2_{I,L}}{NC_L + 1} \; mit \; L \; (L \neq L1; L = 1, 2, ..., K) \tag{5.26}$$

Ist zum Beispiel der Abstand eines Punktes I zu $L3$ kleiner als der zu $L1$, wandert dieser Punkt I zu $L3$. Sowohl $L1$ als auch $L3$ bleiben im lebenden Set. Ist keiner der Abstände kleiner als der zum bestehenden Cluster, so bleibt der Punkt dem ursprünglichen Cluster zugeordnet. Diese Analyse wird für alle Punkte I durchgeführt. Ändert sich die Zugehörigkeit für einen Cluster bei einem Durchlauf über alle Punkte I nicht, so wird er mit all seinen Punkten aus dem lebenden Set genommen. Für die noch lebenden Cluster werden die Mittelpunkte berechnet und das Verfahren erneut durchlaufen. Dieses Vorgehen wird so lange wiederholt, bis sich keine Cluster mehr im lebenden Set befinden (Hartigan & Wong 1979).

k-Means hat an dieser Stelle den Vorteil, dass bis auf die Anzahl an Clustern, keine weiteren Parameter übergeben werden müssen. Um zu überprüfen, ob die Anzahl an Clustern ausreichend ist, wird der CHI verwendet (Caliński & Harabasz 1974). Der CHI berechnet sich anhand von:

$$CHI = (n - NC) \cdot \frac{\sum Diag(B)}{(NC - 1) \cdot \sum Diag(W)} \tag{5.27}$$

wobei n der Anzahl an Sätzen, NC der Anzahl an Clustern, B der Distanz zwischen den Clustermittelwerten und W der Kovarianzmatrix innerhalb der Cluster entspricht. Ziel des Clusterings ist es, eine möglichst saubere Trennung der Datenpunkte zu erreichen. Diese manifestiert sich in einer kleinen Distanz innerhalb und einer großen Distanz zwischen den Clustern. Dementsprechend ist bei einer optimalen Anzahl an Clustern der CHI maximal.

Mithilfe eines maximalen CHI kann somit die optimale Anzahl an Clusterzentren bestimmt werden. Wird diese Anzahl an den k-Means Algorithmus übergeben, können damit die Sätze zu Lesarten geclustert werden. Als Ergebnis wird die Clusternummer zurückgegeben.

4. Darstellung im Frontend

Die Lesartenfunktion verändert den KWIC Ergebnistyp. Im Frontend wird anstatt einer fortlaufenden Zeilennummer nun die Clusterzugehörigkeit angezeigt. Anhand dieser angepassten KWIC können die Nutzer_innen die unterschiedlichen Lesarten des Suchbegriffs inhaltlich evaluieren.

5. Nutzen in der Schreibberatung

Aus Sicht von Schreibberatung nimmt die bisherige Forschung zu Lesarten eher eine $a\ poster-iori$ Sicht ein. Lesarten werden vor allem dazu verwendet, die Komplexität von Texten zu evaluieren (McNamara, Crossley & McCarthy 2010, McNamara, Cai & Louwerse 2007, McNamara et al. 2014, Abba, Joshi & Ji 2019, Wolfe, Widmer, Torrese & Dandignac 2018, Callihan 2017, Dowell, Graesser & Cai 2016, Westerlund 2019, McCarthy, Lightman, Dufty & McNamara 2019) oder aber sie auf ihre Kompatibilität mit einem vorgegebenen Erwartungshorizont zu überprüfen (Shermis & Burstein 2013, Burstein, Tetreault & Madnani 2013, Somasundaran, Burstein & Chodorow 2014, Burstein, Elliot & Molloy 2016, Farra, Somasundaran & Burstein 2015). Der Educational Testing Service (ETS) etwa nutzt solche Verfahren in seinen Anwendungen, um eingereichte Essays mit bereits bestehenden und benoteten Essays zu vergleichen[90].

[90]Der ETS ist eine amerikanische Organisation, die unter anderem den Test of English as a Foreign Language (TOEFL) anbietet. Da dieser Test weltweit als mögliche Zugangsvoraussetzung für ein Hochschulstudium genutzt

Trotz zahlreicher Veröffentlichungen lag der Fokus bisher jedoch nicht darauf, diese Verfahren auch einzusetzten, um Studierende beim Schreiben zu unterstützen.

Didaktische Forschung zum Schreiben in einer Fremdsprache, die sich auf die Verwendung von Synonymen und Homonymen fokussiert, wählt vielfach einen kollokations- oder registerbasierten Ansatz. Vor allem zur Verwendung von Synonymen durch chinesische Lerner des Englischen existiert viel Forschungsmaterial (Zhang & Liu 2005, Jun-Mei 2008, Wang & Ren 2009, Jian-xue 2015). Gemein ist diesen Studien, dass sie von Synonymen ausgehen und versuchen auf Basis von Kollokationen und Registern zu ergründen, welche Variante im entsprechenden Kontext zu wählen ist. In Bezug auf HanConc ergeben sich aus diesem Ansatz drei Schwierigkeiten: In allen Fällen ist das Register festgelegt auf wissenschaftliche Arbeiten. Kollokationsanalysen greifen unabhängig von ihrer statistischen Auswertung zu kurz, da sie nur einen begrenzten Raum um das Suchwort berücksichtigen und die Semantik des umliegenden Kontextes vernachlässigen. Außerdem gehen diese Studien von $a\ priori$ definierten Synonymen aus.

Auf gleiche Weise greifen Schreibratgeber zu kurz, die Schreibenden Listen mit Beispielen als Unterstützung anbieten. Batovski (2008) oder Schroth-Wiechert (2011) beispielsweise zeigen mit ihren Formulierungshilfen Wege auf, die lexikalische Bandbreite des zu schreibenden Textes zu erhöhen. Allerdings sind diese Listen $a\ priori$ festgelegt und nicht $ad\ hoc$ erweiterbar. Außerdem bleiben die Analysen vielfach oberflächlich, sodass Batovski (2008) etwa angibt, dass „security", „protection" und „encryption" bedeutungsgleich verwendet werden können. Während „security" und „protection" im Kontext von Gebäudesicherheit gegebenenfalls so eingesetzt werden können, führen schon einfache Beispiele dazu, dass die These nicht gehalten werden kann. Folgendes Beispiel ist dem BNC entnommen:

1. „The lethal message is that those who have AIDS can engage in sex as long as they use a condom."[91]

„a condom" kann in diesem Zusammenhang mit „protection" ersetzt werden. „Security" hingegen kann vielleicht noch dazu führen, dass der Gesprächspartner das Kommunikationsziel erkennt, „encryption" jedoch wird vor allem in technischen Zusammenhängen und dort in Bezug auf Datenverschlüsselung und IT Sicherheit verwendet. Schreibratgeber können auf Grund des statischen Mediums keine tiefgreifende und fachgebietsspezifische Hilfestellung geben, sondern höchstens eine erste Indikation für mögliche Synonyme liefern.

Die Relevanz der Lesartenfunktion ergibt sich aus der Schwierigkeit Studierender, ihre lexikalischen Probleme zu benennen (Nakamaru 2010). Auch fortgeschrittene Schreibende stellt es vor Herausforderungen Synonyme, Hypernyme und Antonyme zu identifizieren (Liu 2000). Lesarten sind, vor allem in Kombination mit anderen Ergebnisstypen, dafür gedacht, Studierende und Schreibberatende auf mögliche kontextabhängige Bedeutungsunterschiede aufmerksam zu machen. Die Idee ist weniger, automatisch Listen entsprechend derer von Batovski und

wird, liegt es nahe, einen Teil der Bewertung zu automatisieren. Veröffentlichungen des ETS zeigen, dass dies in einem ausreichenden Maße gelingt (Chen, Fife, Bejar & Rupp 2016).

[91] Dieses Beispiel entstammt einem Nachrichtenartikel aus den frühen 1990ern und ist daher inhaltlich veraltet. Es wird an dieser Stelle zu Illustrationszwecken verwendet.

Schroth-Wiechert zu erzeugen, sondern zu weiterführenden datenbasierten Überlegungen anzuregen (Palmquist 2019).

5.3.9 Protokollierung der Nutzer_inneneingaben

Zur technischen und fachlichen Auswertung der Nutzung von HanConc können zwei Logging Methoden genutzt werden. Da HanConc auf R Shiny basiert, gibt die Kommandozeile zur Laufzeit die R Logs aus[92]. Diese können vor allem für technische Belange wie Fehlerbehebungen oder Performanceprobleme eingesetzt werden. Eine inhaltliche Verbesserung von HanConc kann einerseits auf Basis von Beobachtungen der Studierenden oder Schreibberater_innen geschehen oder andererseits über die Benutzer_inneneingaben.

Das Frontend von HanConc ist mit einem Startknopf versehen, welcher dafür sorgt, dass Benutzer_inneneingaben erst dann verarbeitet werden, wenn die Eingabe abgeschlossen ist. Wird der Startknopf betätigt, beginnt HanConc mit der Berechnung der Ergebnisse und speichert gleichzeitig die Benutzer_inneneingaben in einer Protokolldatei ab. Hierzu werden die Eingaben zu einem DataFrame zusammengefasst und dieses in eine CSV Datei auf die Festplatte geschrieben. Es ist zu bedenken, dass bewusst keine Informationen wie Session ID oder Uhrzeiten gespeichert werden, um Diskussionen zum Datenschutz zu vermeiden.

Zum Zeitpunkt des Verfassens dieser Arbeit sind erst wenige Studierende mit HanConc betreut worden. Auf eine Auswertung der bisherigen Ergebnisse wird daher mit Hinblick auf statistische Hindernisse verzichtet.

5.4 Fazit zu HanConc

HanConc wurde im Rahmen eines Forschungsprojekts der LUH für den Einsatz an Schreibzentren entwickelt. Da von Anfang an nur begrenzte Entwicklungsmöglichkeiten und Zeit zur Verfügung standen, wurde HanConc so programmiert, dass es unabhängig von den ursprünglich handelnden Personen betrieben und weiterentwickelt werden kann.

Grundsätzlich wurde HanConc mit Blick auf ein möglichst breites Anwendungsspektrum programmiert. Schreibberater_innen soll es möglich sein, die Software mit Studierenden in einem Seminarraum auf deren Laptops zu nutzen oder sie auf einem Server in der Cloud zu installieren und gleichzeitig hunderten Nutzer_innen zur Verfügung zu stellen. Mit der hier präsentierten Architektur ist beides problemlos möglich. HanConc ist in seiner Infrastruktur- und Software-Architektur darauf ausgerichtet, einen Startpunkt für Anpassungen und Erweiterungen zu liefern. Open Source ist hier nicht nur als eine Art der Bereitstellung von Software sondern als Aufforderung zur Weiterentwicklung gemeint.

HanConcs Funktionen vereinen Forschungen verschiedener Disziplinen, die Schreibberater_innen und Studierende dabei unterstützen können, nutzbare Erkenntnisse aus Korpora zu

[92] Wird R über eine Kommandozeile ausgeführt, d.h. über CMD oder PowerShell unter Windows oder eine Linux Shell, so kann diese dazu genutzt werden, die Ausgabe von R in eine Log-Datei umzuleiten. Diese Funktion ist jedoch nicht implementiert.

gewinnen. Entsprechend der Grundausrichtung von HanConc sind alle Datenmodelle darauf ausgelegt, es interessierten Nutzer_innen zu ermöglichen, diese zu durchsuchen, weiterzuverarbeiten oder an die eigenen Ansprüche anzupassen. Gleichzeitig sorgt die Aufbereitung der Ursprungstexte in diesen Datenmodellen dafür, dass HanConc auch auf schwacher Hardware schnell Ergebnisse liefert. Als Datengrundlage stehen hierfür neben den annotierten Texten auch DTM und Wortlisten zur Verfügung, um erweiterte Funktionen zu ermöglichen.

Die Ergebnistypen von HanConc basieren auf den Traditionen von Schreibberatung, Korpus- und Computerlinguistik und Statistik. Neben klassischen KWIC als Grundfunktion jeder Korpussoftware wird daher auch die unmittelbare Umgebung der Suchwörter betrachtet, ihre Verteilung im einzelnen Text und im gesamten Korpus ausgewertet, der latente Kontext aufbereitet und schlussendlich die Ergebnisse visualisiert. Somit bieten HanConcs Funktionen den fachlichen Startpunkt für zusätzliche Erweiterungen aus unterschiedlichen Disziplinen.

HanConc bemüht sich, Nutzer_innen unterschiedlicher Fachrichtungen einen bekannten Einstiegspunkt zu liefern und als didaktisches Werkzeug zu fungieren. Der Lernerfolg soll dabei sowohl durch die Untersuchung von Fachwissenschaftssprache als auch durch das Kennenlernen von Methoden und Techniken aus anderen auf quantitative Sprachanalyse fokussierten Fachrichtungen erreicht werden. Die in diesem Kapitel aufgezeigten Methoden entstammen dementsprechend auch der fachwissenschaftlichen Forschung und wurden weniger mit Blick auf didaktische Anforderungen einer Schreibberatung entwickelt. Daher können die hier vorgestellten Anwendungsmöglichkeiten in einer Schreibberatung auch nur der Anfang von weiteren didaktischen Überlegungen sein.

Kapitel 6

Schlussbemerkungen

Sprache ist Teil des Studiums und damit auch jeder Schreibberatung

Schreibberater_innen und ihre Studierenden haben unterschiedliche akademische Hintergründe. Die Traditionen einzelner Fachrichtungen und Fakultäten und die damit einhergehenden Schreibstile sind oftmals so verschieden, dass es zwangsläufig dazu kommt, dass Schreibberater_innen aufgrund fehlender sprachlicher und inhaltlicher Kenntnisse Studierenden nicht die Unterstützung anbieten können, die diese bräuchten. Das inhaltliche Wissen um einen Fachbegriff ist folglich kein Garant, ihn in einer wissenschaftlichen Arbeit auch korrekt zu verwenden. Ebenso ist es unzureichend, wenn Schreibberatung Studierenden, die in einer anderen als ihrer Erstsprache schreiben, nur generische Hinweise zum akademischen Schreiben anbietet und sprachliche Aspekte ausklammert.

Im Gegensatz zu den Geisteswissenschaften verlangen die technischen und naturwissenschaftlichen Fächer und Fakultäten akademisches Schreiben erst spät im Studium. Eine effektive Unterstützung ist daher besonders wichtig, um die gegebenenfalls fehlende Erfahrung im akademischen Schreiben auszugleichen. Korpuslinguistik kann einen Beitrag dazu leisten, der Schreibberatung die notwendigen sprachlichen Analysen zur Verfügung zu stellen, um Studierende auch außerhalb der inhaltlichen Expertise der Schreibberater_innen unterstützen zu können. Mit dieser Arbeit, dem eigens erstellten Korpus und der dazugehörigen Software ist die Lücke zwischen den Erkenntnissen der quantitativen Sprachforschung und den Herausforderungen von Schreibberatung geschlossen worden.

Dissertationen sind eine geeignete Grundlage für ein universitäres Schreibberatungskorpus

Der Hannover Concordancer (HanConc) ist ein korpusgestütztes Tool zur fortgeschrittenen Analyse akademischer Texte, welches insbesondere für die Unterstützung von Schreibberatungen entwickelt wurde. Das Hannover Advanced Academic Writing Corpus (HAAWC) hat sich als wertvolle Ressource für die Entwicklung von HanConc und die Überlegungen zu Schreibberatungen erwiesen. Obwohl für die Erstellung des Korpus nur die Dissertationen einer Universität verwendet wurden, ist die Textgrundlage ausreichend groß, um sinnvoll mit ihr arbeiten

zu können. Mittels Terminology Extraction und Machine Learning konnte gezeigt werden, dass sich die Einteilung des Korpus nach Fakultäten nachweisbar auch in den Texten wiederfindet. Diese Einteilung ist für Schreibberatungen notwendig, da die sprachlichen Unterschiede zwischen den einzelnen Fakultäten deutlich über das Fachvokabular hinausgehen. Damit ist gezeigt worden, dass es keine Schreibberatung geben sollte, die auf diese Unterschiede keine Rücksicht nimmt. Das HAAWC liefert Schreibberater_innen somit die Möglichkeit, gezielter auf die sprachlichen Bedürfnisse der Studierenden einzugehen.

Schreibberatung braucht eigene Korpussoftware

Die Anforderungen eines Schreibzentrums an eine Korpussoftware weichen von denen der linguistischen Forschung ab. Schreibberatung braucht die Möglichkeit eigene Texte zu analysieren, um sich auf die Bedürfnisse der zu beratenden Studierenden einstellen zu können. Außerdem muss die Software gleichzeitig einfach zu bedienen und umfassend in ihren Funktionen sein. HanConc setzt hier an, indem seine Funktionen in ihrer Komplexität ansteigen und damit Nutzer_innen die Möglichkeit geben, entsprechend ihrer Vorkenntnisse Texte analysieren zu lassen. Ziel von HanConc ist es, dass Nutzer_innen selbst beim Erstkontakt mit Korpuslinguistik innerhalb kurzer Zeit nutzbare Analysen und Ergebnisse erzeugen können. Dennoch wurden keine Abstriche bei den Algorithmen und Verfahren gemacht und Erkenntnisse der Korpus- und Computerlinguistik, der Statistik und der Informatik implementiert. Diese Funktionen sollten ausreichen, um mit HanConc Forschungsergebnisse zu generieren, um damit am aktuellen akademischen Diskurs teilnehmen zu können.

Mittels einer Umfrage konnte überprüft werden, ob die Annahmen, die für den Einsatz von Korpuslinguistik in Schreibberatungen getroffen wurden, auch von anderen Schreibberater_innen bestätigt werden. 65 Schreibberater_innen haben Auskunft zu ihrer eigenen akademischen Vorgeschichte, zu ihrem Schreibzentrum, ihren Studierenden, ihren Schreibberatungen und ihren Herausforderungen beim Einsatz von Korpuslinguistik in ihrer täglichen Arbeit gegeben. Die Ergebnisse zeigen, dass sich sowohl die Erfahrungen des Schreibzentrums der Leibniz Universität Hannover (LUH) auf andere Schreibzentren übertragen lassen als auch die Annahmen zum Einsatz von Korpora in der Schreibberatung zutreffen. Die Umfrage hat zudem ergeben, dass etwa 25% der Schreibberater_innen HanConc oder andere Korpussoftware bereits nutzen oder der Idee gegenüber, Korpuslinguistik in ihrer Schreibberatung einzusetzen, aufgeschlossen sind. Weitere 25% haben keine grundsätzlichen Bedenken gegenüber Korpuslinguistik in der Schreibberatung, benötigen aber mehr Informationen und Schulungen, um sinnvoll beurteilen zu können, ob HanConc eine wertvolle Ressource für sie wäre.

HanConc nimmt die Idee von Open Source Software ernst, denn der Quellcode wird mit dieser Arbeit veröffentlicht und wurde zusätzlich so designt, dass er möglichst einfach zu verstehen und zu bearbeiten ist. Damit gibt die Software auch Interessierten, die wenig Programmiererfahrung haben, die Möglichkeit, ihre eigene Version des Programms zu gestalten. Durch die gewählte Architektur kann HanConc sowohl auf einzelnen Computern für einzelne Schreibberater_innen als auch auf einem Server und damit für ganze Schreibzentren eingesetzt werden.

Erst die Kombination aus einem Korpus bestehend aus spezifischen und hochqualitativen Texten und einem Werkzeug, dass speziell für diesen Einsatzzweck programmiert wurde, kann der Korpuslinguistik als unterstützende Methode bei Schreibberatungen zu einer größeren Verbreitung verholfen werden. Durch die offene Programmierung und die einfache Integrierbarkeit neuer Texte kann auch in Zukunft sichergestellt werden, dass HanConc sich an die Anforderungen moderner Schreibzentren anpasst.

HanConc kann dabei helfen, dass durch Schreibberatung mehr Studierende erfolgreich schreiben

Schreibzentren haben den Auftrag, Studierenden beim Verfassen ihrer akademischen Arbeiten zu helfen. Um ihnen die bestmögliche Schreibberatung zur Verfügung zu stellen, könnte die Anzahl an Schreibberater_innen mit unterschiedlichen akademischen Hintergründen erhöht werden. Andererseits könnte der Einsatz von Werkzeugen wie HanConc erwogen werden, um den oben beschriebenen Herausforderungen zu begegnen. Da es kostenlos, anpassbar, einfach zu bedienen und auch auf vorhandener Hardware lauffähig ist, ist HanConc die praktikablere, günstigere und damit bessere Alternative. Es ist zu hoffen, dass Korpuslinguistik und HanConc vermehrt eingesetzt werden, um Schreibberater_innen dabei zu unterstützen, bessere und zielführendere Beratungen anzubieten.

Literaturverzeichnis

Abba, K. A., Joshi, R. M. & Ji, X. R. (2019), 'Analyzing writing performance of l1, l2, and generation 1.5 community college students through coh-metrix', *Written Language & Literacy* **22**(1), 67–94.

Agrawal, R., Srikant, R. et al. (1994), Fast algorithms for mining association rules, *in* 'Proceedings of the 20th Very Large Databases Conference', Vol. 1215, pp. 487–499.

Aitchison, J. (2003), *The Words in Mind - An Introduction to the Mental Lexicon*, Blackwell Publishers Inc.

Alghamdi, R. & Alfalqi, K. (2015), 'A survey of topic modeling in text mining', *International Journal of Advanced Computer Science and Applications (IJACSA)* **6**(1).

Alrehamy, H. H. & Walker, C. (2017), Semcluster: unsupervised automatic keyphrase extraction using affinity propagation, *in* 'UK Workshop on Computational Intelligence', Springer, pp. 222–235.

Altenberg, B. & Granger, S. (2001), 'The grammatical and lexical patterning of MAKE in native and non-native student writing', *Applied Linguistics* **22**(2), 173–194.

Amjadian, E., Inkpen, D., Paribakht, T. & Faez, F. (2016), Local-global vectors to improve unigram terminology extraction, *in* 'Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)', The COLING 2016 Organizing Committee, pp. 2–11.

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S. & Collins, M. (2016), 'Globally normalized transition-based neural networks', *arXiv* .

Anscombe, F. (1949), 'The statistical analysis of insect counts based on the negative binomial distribution', *Biometrics* **5**(2), 165–173.

Anthony, L. (2019), 'Antconc (3.5.8)'.

Artetxe, M., Labaka, G. & Agirre, E. (2018), 'Unsupervised statistical machine translation', *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* .

Attali, Y. & Burstein, J. (2006), 'Automated essay scoring with e-rater v. 2', *The Journal of Technology, Learning and Assessment* **4**(3).

Baayen, R. (2001), *Word frequency distributions*, Kluwer Academic Publishing, Dordrech.

Baayen, R. (2008), *Analyzing linguistic data: A practical introduction to statistics using R*, Cambridge University Press.

Ballweg, S. (2011), 'Schreibberatung für internationale studierende', *Sprachlernberatung für DaF* pp. 123–136.

Batovski, D. (2008), 'How to use technical synonyms and antonyms', *Asummption University Journal of Technology* **12**(2), i–ii.

Belz, J. A. (2004), 'Learner corpus analysis and the development of foreign language proficiency', *System* **32**(4), 577–591.

Bentz, C., Kiela, D., Hill, F. & Buttery, P. (2014), 'Zipf's law and the grammar of languages: A quantitative study of Old and Modern English parallel texts', *Corpus Linguistics and Linguistic Theory* .

Bergamaschi, S. & Po, L. (2014), Comparing lda and lsa topic models for content-based movie recommendation systems, *in* 'International Conference on Web Information Systems and Technologies', Springer, pp. 247–263.

Biber, D. (1992), 'On the complexity of discourse complexity: A multidimensional analysis', *Discourse Processes* **15**(2), 133–163.

Biber, D. (1993), 'Co-occurrence patterns among collocations: a tool for corpus-based lexical knowledge acquisition', *Computational Linguistics* **19**(3), 531–538.

Bird, S., Klein, E. & Loper, E. (2009), *Natural Language Processing with Python*, O'Reilly Media.

Bortz, J. (2010), *Statistik für Human- und Sozialwissenschaftler*, Springer.

Boulton, A. (2010), *Learning outcomes from corpus consultation*, Equinox.

Bower, J. & Kawaguchi, S. (2011), 'Negotiation of meaning and corrective feedback in Japanese/English eTandem', *Language Learning & Technology* **15**(1), 41–71.

Bräuer, G. (2006), Schüler helfen Schülern–Schreibberatung in der Schule, *in* 'Forum Schulstiftung', Vol. 45, pp. 24–37.

Briscoe, T. (2007), 'Language learning, power laws, and sexual selection', *Mind & Society* **7**(1), 65–76.

Bundesamt für Sicherheit in der Informationstechnik (2020), *IT-Grundschutz-Kompendium*, Bundesanzeiger Verlag, Köln.

Burstein, J. & Chodorow, M. (2010), Progress and new directions in technology for automated essay evaluation, *in* R. Kaplan, ed., 'The Oxford handbook of applied linguistics', Oxford University Press.

Burstein, J., Elliot, N. & Molloy, H. (2016), 'Informing automated writing evaluation using the lens of genre: Two studies.', *Calico Journal* **33**(1), 117–141.

Burstein, J., Tetreault, J. & Madnani, N. (2013), The e-rater automated essay scoring system, *in* M. D. Shermis & J. Burstein, eds, 'Handbook of automated essay evaluation', Routledge, pp. 77–89.

Bußmann, H. & Lauffer, H. (2008), *Lexikon der Sprachwissenschaft*, Kröner Stuttgart.

Caliński, T. & Harabasz, J. (1974), 'A dendrite method for cluster analysis', *Communications in statistics-theory and methods* **3**(1), 1–27.

Callihan, R. (2017), 'A comparison of discourse connective identification of coh-metrix and the penn discourse treebank'.

Calwer Verlag (1979), *Große Konkordanz zur Lutherbibel*, Calwer Verlag, Stuttgart.

Cambria, E., Schuller, B., Xia, Y. & Havasi, C. (2013), 'New Avenues in Opinion Mining and Sentiment Analysis', *IEEE intelligent systems* **28**(2), 15–21.

Cameron, A. C. & Trivedi, P. K. (2005), *Microeconometrics: Methods and Applications*, Cambridge University Press.

Cameron, a. C. & Trivedi, P. K. (2013), 'Count Panel Data', *Oxford Handbook of Panel Data Econometrics* .

Chakrabarti, K., Chaudhuri, S., Cheng, T. & Xin, D. (2012), A framework for robust discovery of entity synonyms, *in* 'Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining', ACM, pp. 1384–1392.

Chan, T.-P. & Liou, H.-C. (2005), 'Effects of web-based concordancing instruction on EFL students' learning of verb–noun collocations', *Computer Assisted Language Learning* **18**(3), 231–251.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L. & Blei, D. M. (2009), Reading tea leaves: How humans interpret topic models, *in* 'Advances in neural information processing systems', pp. 288–296.

Chang, J.-S. & Chang, Y.-C. (2004), Computer assisted language learning based on corpora and natural language processing: the experience of project candle, *in* 'An Interactive Workshop on Language e-Learning', pp. 15–23.

Chang, Y.-C., Chang, J. S., Chen, H.-J. & Liou, H.-C. (2008), 'An automatic collocation writing assistant for Taiwanese EFL learners: A case of corpus-based NLP technology', *Computer Assisted Language Learning* **21**(3), 283–299.

Chen, J., Fife, J., Bejar, I. & Rupp, A. (2016), 'Building e-rater scoring models using machine learning methods', *ETS Research Report Series* **2016**(1), 1–12.

Cheng, W., Greaves, C. & Warren, M. (2006), 'From n-gram to skipgram to concgram', *International journal of corpus linguistics* **11**(4), 411–433.

Chitez, M., Rapp, C. & Kruse, O. (2015), 'Corpus-supported academic writing: how can technology help?', *Critical CALL – Proceedings of the 2015 EUROCALL Conference, Padova, Italy* (2015), 125–132.

Chiu, J. P. & Nichols, E. (2016), 'Named entity recognition with bidirectional lstm-cnns', *Transactions of the Association for Computational Linguistics* **4**, 357–370.

Chowdhury, F. M., Gliozzo, A. M. & Trewin, S. M. (2018), 'Domain-specific terminology extraction by boosting frequency metrics'. US Patent App. 15/469,766.

Christ, O. (1994), 'The IMS corpus workbench technical manual', *Institut für maschinelle Sprachverarbeitung, Universität Stuttgart* .

Christ, O., Schulze, B. M., Hofmann, A. & Koenig, E. (1999), 'The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual', *Institut für maschinelle Sprachverarbeitung, Universität Stuttgart* .

Cuéllar, M. T. A. (2013), 'Process writing and the development of grammatical competence', *HOW Journal* **20**(1), 11–35.

Cutting, D. & Pedersen, J. (1989), Optimization for dynamic inverted index maintenance, *in* 'Proceedings of the 13th annual international ACM SIGIR conference on research and development in information retrieval', pp. 405–411.

D'agostino, R. B., Belanger, A. & D'Agostino Jr, R. B. (1990), 'A suggestion for using powerful and informative tests of normality', *The American Statistician* **44**(4), 316–321.

Daskalovska, N. (2015), 'Corpus-based versus traditional learning of collocations', *Computer Assisted Language Learning* **28**(2), 130–144.

Dawood, H. (2014), 'The impact of immediate grammatical error correction on senior english majors' accuracy at hebron university', *International Journal of Foreign Language Teaching and Research* **2**(7), 37–46.

De Haan, P. & Van der Haagen, M. (2013), 'The search for sophisticated language in advanced EFL writing: A longitudinal study', *Dutch Journal of Applied Linguistics* **2**(1), 16–27.

De Saussure, F. (2011), *Course in General Linguistics*, Columbia University Press.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American society for information science* **41**(6), 391–407.

Delignette-Muller, M. L. & Dutang, C. (2015), 'fitdistrplus: An R package for fitting distributions', *Journal of Statistical Software* **64**(4), 1–34.

Dengscherz, S. (2020), 'Professionelles Schreiben in mehreren Sprachen–das PROSIMS-Schreibprozessmodell', *Zeitschrift für Interkulturellen Fremdsprachenunterricht* **25**(1).

Derczynski, L., Maynard, D., Rizzo, G., Van Erp, M., Gorrell, G., Troncy, R., Petrak, J. & Bontcheva, K. (2015), 'Analysis of named entity recognition and linking for tweets', *Information Processing & Management* **51**(2), 32–49.

Diab, N. M. (2015), 'Effectiveness of written corrective feedback: Does type of error and type of correction matter?', *Assessing Writing* **24**, 16–34.

Disterer, G. (2013), 'ISO/IEC 27000, 27001 and 27002 for Information Security Management'.

Dittmann, J., Geneuss, K. A., Nennstiel, C. & Quast, N. A. (2003), Schreibprobleme im Studium–eine empirische Untersuchung, *in* K. Ehlich & A. Steets, eds, 'Wissenschaftlich schreiben–lehren und lernen', de Gruyter, pp. 155–185.

Django Software Foundation (n.d.), 'Django'.
URL: *https://djangoproject.com*

Dowell, N. M., Graesser, A. C. & Cai, Z. (2016), 'Language and discourse analysis with Coh-Metrix: Applications from educational material to learning environments at scale', *Journal of Learning Analytics* **3**(3), 72–95.

Dudenredaktion (2006), *Der Duden in 12 Banden: 1 - Die deutsche Rechtschreibung*, Dudenverlag, Mannheim.

Durrant, P. L. (2010), *High frequency collocations and second language learning*, Nottingham University Press.

Durrant, P. & Schmitt, N. (2009), 'To what extent do native and non-native writers make use of collocations?', *IRAL - International Review of Applied Linguistics in Language Teaching* **47**(2), 157–177.

Einig, C. & Menne-El Sawy, G. (2012), 'Problemfeld: Sprachliche Register in der Wissenschaftssprache', *Informationen Deutsch als Fremdsprache* **39**(4), 385–404.

Ekštein, K. & Krčmář, L. (2013), Automatic LSA-based retrieval of synonyms (for search space extension), *in* 'Recent Progress in Data Engineering and Internet Technology', Springer, pp. 79–85.

Ellis, N. (2012), 'Formulaic language and second language acquisition: Zipf and the phrasal teddy bear', *Annual Review of Applied Linguistics* **32**, 17–44.

Estrada, R. & Ruiz, I. (2016), 'Big Data Smack', *Apress, Berkeley, CA* .

Europäische Kommision (2015), 'ECTS Leitfaden'.
URL: *https://ec.europa.eu/education/ects/users-guide/docs/ects-users-guide_de.pdf*

Evert, S. (2005), *The Statistics of Word Cooccurrences Word Pairs and Collocations – Unveröffentlichte Promotion am Institut für maschinelle Sprachverarbeitung der Universität Stuttgart*, Publication Server of the University of Stuttgart.
URL: *http://en.scientificcommons.org/19948039*

Evert, S. & Baroni, M. (2007), 'zipfR: Word frequency distributions in R', *Proceedings of the 45th Annual Meeting of the ACL* (V).

Evert, S. & Hardie, A. (2011), Twenty-first century corpus workbench: Updating a query architecture for the new millennium, *in* 'Proceedings of the Corpus Linguistics 2011 Conference'.

Fan, W., Wang, X., Wu, Y. & Xu, J. (2015), 'Association rules with graph patterns', *Proceedings of the VLDB Endowment* **8**(12), 1502–1513.

Farra, N., Somasundaran, S. & Burstein, J. (2015), Scoring persuasive essays using opinions and their targets, *in* 'Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications', pp. 64–74.

Feinerer, I., Hornik, K. & Meyer, D. (2008), 'Text Mining Infrastructure in R', *Journal of Statistical Software* **25**(5), 1–54.

Fellows, I. (2018), *wordcloud: Word Clouds*. R package version 2.6.

Francis, L. & Flynn, M. (2010), 'Text Mining Handbook', *Casualty Actuarial Society E-Forum* pp. 1–61.
URL: *http://www.casact.net/pubs/forum/10spforum/CompleteS10.pdf#page=5*

Friginal, E. & Weigle, S. (2014), 'Exploring multiple profiles of L2 writing using multidimensional analysis', *Journal of Second Language Writing* pp. 1–16.

Furnas, G. W., Landauer, T. K., Gomez, L. M. & Dumais, S. T. (1987), 'The vocabulary problem in human-system communication', *Communications of the ACM* **30**(11), 964–971.

Girgensohn, K. & Peters, N. (2012), '"At University nothing speaks louder than research" Plädoyer für Schreibzentrumsforschung', *Zeitschrift Schreiben* (2009).

Granger, S. (1996), 'From CA to CIA and back: An integrated approach to computerized bilingual and learner corpora', *Languages in Contrast. Textbased cross-linguistic studies* pp. 37–51.

Granger, S. (2011), 'From phraseology to pedagogy: Challenges and prospects', *Chunks in the Description of Language. A tribute to ...* pp. 123–146.

Granger, S. & Tyson, S. (1996), 'Connector usage in the English essay writing of native and non-native EFL speakers of English', *World Englishes* **15**(1), 17–27.

Gries, S. (2015), 'Some Current Quantitative Problems in Corpus Linguistics and a Sketch of Some Solutions', *Language and Linguistics* **16**, 93–117.

Gries, S. T. (2009), *Statistics for Linguistics with R: A Practical Introduction*, De Gruyter Mouton.

Grieshammer, E. (2011), *Der Schreibprozess beim wissenschaftlichen Schreiben in der Fremdsprache Deutsch und Möglichkeiten seiner Unterstützung*.
URL: *https://opus4.kobv.de/opus4-euv/frontdoor/index/index/year/2011/docId/49*

Grieshammer, E. (2013), *Zukunftsmodell Schreibberatung : eine Anleitung zur Begleitung von Schreibenden im Studium*, Schneider, Hohengehren, Baltmannsweiler.

Grotjahn, R. (2002), Konstruktion und Einsatz von C-Tests: Ein Leitfaden für die Praxis, *in* R. Grotjahn, ed., 'Der C-Test. Theoretische Grundlagen und praktische Anwendungen', AKS-Verlag, Bochum, pp. 211–225.

Gärtner, T. (2013), Amplifier Collocation in Native and Non-Native Speaker Language Performance, *in* A. Ammermann, B. Alexander, J. Pflaeging & P. Schildhauer, eds, 'Facets of Linguistics: Proceedings of the 14th Norddeutsches Linguistisches Kolloquium 2013 in Halle (Saale)', Peter Lang, Frankfurt am Main, pp. 23–34.

Gärtner, T. (2014), *Applying multivariate estimation models to investigate factors promoting the use of the passive voice in the writing of EFL learners*.
URL: *https://www.researchgate.net/publication/368576546_Applying_multivariate_estimation_models_to_investigate_factors_promoting_the_use_of_the_passive_voice_in_the_writing_of_EFL_learners*

Ha, T.-L., Cho, E., Niehues, J., Mediani, M., Sperber, M., Allauzen, A. & Waibel, A. (2016), The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2017, *in* 'Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers', Vol. 2, pp. 303–310.

Halliday, M. A. K. & Hasan, R. (2014), *Cohesion in English*, Routledge.

Han, Y. & Hyland, F. (2015), 'Exploring learner engagement with written corrective feedback in a Chinese tertiary EFL classroom', *Journal of Second Language Writing* **30**, 31–44.

Hartigan, J. A. & Wong, M. A. (1979), 'Algorithm AS 136: A k-means clustering algorithm', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1), 100–108.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The elements of Statistical Learning*, Springer.

Hauge, M. V., Stevenson, M. D., Rossmo, D. K. & Le, S. C. (2016), 'Tagging Banksy: using geographic profiling to investigate a modern art mystery', *Journal of Spatial Science* **61**(May), 185–190.

Hayes, J. (1996), 'A New Framework for Understanding Cognition and Affect in Writing', *The Science of Writing. Theories, Methods, Individual Differences, and Applications* pp. 1–27.

He, W., He, Z., Wu, H. & Wang, H. (2016), Improved neural machine translation with SMT features, *in* 'Thirtieth AAAI conference on Artificial Intelligence'.

Heift, T. & Schulze, M. (2007), *Errors and intelligence in computer-assisted language learning: Parsers and pedagogues*, Routledge.

Hempelmann, C. F., Dufty, D., McCarthy, P. M., Graesser, A. C., Cai, Z. & McNamara, D. S. (2005), Using LSA to automatically identify givenness and newness of noun phrases in written discourse, *in* 'Proceedings of the 27th annual Conference of the Cognitive Science Society', Erlbaum Mahwah, NJ, pp. 941–946.

Hennig, C. (2019), *fpc: Flexible Procedures for Clustering*. R package version 2.2-3.

Hirschberg, D. (1997), *Serial Computations of Levenshtein Distances*, Oxford University, p. 123–141.

Hofmann, T. (2000), Learning the similarity of documents: An information-geometric approach to document retrieval and categorization, *in* 'Advances in Neural Information Processing Systems', pp. 914–920.

Hofmann, T. (2001), 'Unsupervised Learning by Probabilistic Latent Semantic Analysis', *Machine Learning* pp. 177–196.

Homburg, C. & Krohmer, H. (2009), *Marketingmanagement - Strategie - Instrumente - Umsetzung - Unternehmensführung*, 3. überarb. u. erw. aufl. 2009 edn, Gabler, Wiesbaden.

Homburg, C. & Schäfer, H. (2002), 'Die erschließung von kundenpotenzialen durch cross-selling: Konzeptionelle grundlagen und empirische ergebnisse', *Marketing ZFP* **24**(1), 7–26.

Hommerberg, C. & Tottie, G. (2007), 'Try to or try and? Verb complementation in British and American English', *ICAME Journal: Computers in English ...* **31**, 45–64.

Hornik, K. & Grün, B. (2011), 'topicmodels: An r package for fitting topic models', *Journal of statistical software* **40**(13), 1–30.

Hsueh, S.-C. & Kuo, C.-H. (2017), Effective Matching for P2P Lending by Mining Strong Association Rules, *in* 'Proceedings of the 3rd International Conference on Industrial and Business Engineering', ACM, pp. 30–33.

Ishikawa, S. (2009), 'Phraseology overused and underused by Japanese learners of English: A contrastive interlanguage analysis', *Phraseology, corpus linguistics and lexicography: ...* pp. 87–100.

J. R. Hayes, L. S. F. (1980), 'Identifying the Organization of Writing Processes', *Cognitive Processes in Writing* pp. 3–30.

Jafarpour, A. A. & Sharifi, A. (2012), 'The effect of error correction feedback on the collocation competence of Iranian EFL learners', *Teaching English with Technology* **12**(3), 3–17.

Jarque, C. M. & Bera, A. K. (1980), 'Efficient tests for normality, homoscedasticity and serial independence of regression residuals', *Economics Letters* **6**(3), 255–259.

Ji, S., Li, G., Li, C. & Feng, J. (2009), Efficient interactive fuzzy keyword search, *in* 'Proceedings of the 18th International conference on World Wide Web', ACM, pp. 371–380.

Jian-xue, M. (2015), 'Corpus-based Differentiation of English Synonyms——Taking Offer, Provide and Supply as an Example', *Education Modernization* (14), 44.

Jun-Mei, L. (2008), 'A Corpus-based Research on Chinese EFL Learners' Synonyms Learning', *Journal of Sichuan College of Education* **11**.

Karan, M., Šnajder, J. & Bašic, B. D. (2012), 'Distributional semantics approach to detecting synonyms in Croatian language', *Information Society* pp. 111–116.

Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S. & Finin, T. (2016), 'Robust semantic text similarity using LSA, machine learning, and linguistic resources', *Language Resources and Evaluation* **50**(1), 125–161.

Katz, G. & Giesbrecht, E. (2006), 'Automatic identification of non-compositional multi-word expressions using latent semantic analysis', ... *Workshop on Multiword Expressions: Identifying* ... (July), 12–19.

Kemps-Snijders, M., Brouwer, M., Kunst, J. P. & Visser, T. (2012), Dynamic web service deployment in a cloud environment, *in* 'Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)', European Language Resources Association (ELRA), pp. 2941–2944.

Kennedy, C. & Miceli, T. (2010), 'Corpus-assisted creative writing: Introducing intermediate Italian learners to a corpus as a reference resource', *Language Learning & Technology* **14**(1), 28–44.

Kennedy, G. (2003), 'Amplifier collocations in the British National Corpus: Implications for English language teaching', *TESOL Quarterly* **37**, 467.

Kilgarriff, A. (2012), Getting to know your corpus, *in* 'International Conference on Text, Speech and Dialogue', Springer, pp. 3–15.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlỳ, P. & Suchomel, V. (2014), 'The Sketch Engine: ten years on', *Lexicography* **1**(1), 7–36.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. & Rychlý, P. (2008), Gdex: Automatically finding good dictionary examples in a corpus, *in* J. D. Elisenda Bernal, ed., 'Proceedings of the 13th EURALEX International Congress', Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra, Barcelona, Spain, pp. 425–432.

Kilgarriff, A., Rychly, P., Smrz, P. & Tugwell, D. (2004), 'ITRI-04-08 the Sketch Engine', *Information Technology* **105**, 116.

Kirby, G. (1985), 'Zipf's Law', *Journal of Naval Science* **10**(3), 180–185.

Kohn, K. (2009), 'Computer assisted foreign language learning', *Foreign Language Communication and Learning. Handbooks of Applied Linguistics* **6**, 573–603.

Kosem, I., Husak, M. & McCarthy, D. (2011), 'Gdex for slovene', *Proceedings of eLex* pp. 151–159.

Kruse, O., Jakobs, E.-M. & Ruhmann, G. (1999), *Schlüsselkompetenz Schreiben: Konzepte, Methoden, Projekte für Schreibberatung und Schreibdidaktik an der Hochschule*, Luchterhand.

Kunkel-Razum, K. (2006), *Der Duden in 12 Banden: 4 - Die Grammatik*, Dudenverlag, Mannheim.

Leacock, C., Chodorow, M., Gamon, M. & Tetreault, J. (2014), 'Automated grammatical error detection for language learners', *Synthesis Lectures on Human Language Technologies* **7**(1), 1–170.

Lin, D. (1998), Automatic retrieval and clustering of similar words, *in* '36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2', pp. 768–774.

Lin, J. & Dyer, C. (2010), 'Data-Intensive Text Processing with MapReduce', *Synthesis Lectures on Human Language Technologies* **3**, 1–177.

Liu, D. (2000), Writing cohesion: Using content lexical ties in ESOL, *in* 'English Teaching Forum', Vol. 38, pp. 28–33.

Louwerse, M. M., Mccarthy, P. M., Mcnamara, D. S. & Graesser, A. C. (2003), 'Variation in Language and Cohesion across Written and Spoken Registers', *Proceedings of the 26th Annual Meeting of the Cognitive Science Society* (1988).

Luo, Q. & Liao, Y. (2015), 'Using corpora for error correction in EFL learners' writing', *Journal of Language Teaching and Research* **6**(6), 1333–1342.

Luong, M.-T., Pham, H. & Manning, C. D. (2015), 'Effective approaches to attention-based neural machine translation', *arXiv* .

Mahlow, C., Grün, C., Holupirek, A. & Scholl, M. H. (2012), 'A framework for retrieval and annotation in digital humanities using XQuery full text and update in BaseX', *Proceedings of the 2012 ACM symposium on Document Engineering - DocEng '12* p. 195.

Mandelbrot, B. (1965), Information Theory and Psycholinguistics: A Theory of Word Frequencies, *in* B. Wolman & E. Nagel, eds, 'Scientific Psychology', Basic Books, pp. 550–563.

Manning, C. & Schutze, H. (1999), *Foundations of Statistical Natural Language Processing*, Massachusetts Institute of Technology, Boston.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J. & McClosky, D. (2014), The Stanford CoreNLP Natural Language Processing Toolkit, *in* 'Association for Computational Linguistics (ACL) System Demonstrations', pp. 55–60.
URL: *http://www.aclweb.org/anthology/P/P14/P14-5010*

Matsuda, P. K. (1999), 'Composition studies and ESL writing: A disciplinary division of labor', *College composition and communication* **50**(4), 699–721.

McCandless, M., Hatcher, E. & Gospodnetić, O. (2010), *Lucene in Action*, Manning, Birmingham.

McCarthy, P. M., Briner, S. W., Rus, V. & McNamara, D. S. (2007), 'Textual signatures: Identifying text-types using Latent Semantic Analysis to measure the cohesion of text structures', *Natural Language Processing and Text Mining* pp. 107–122.

McCarthy, P. M. & Jarvis, S. (2010), 'MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment', *Behavior Research Methods* **42**(2), 381–392.

McCarthy, P. M., Lightman, E. J., Dufty, D. F. & McNamara, D. S. (2019), Using Coh-Metrix to assess cohesion and difficulty in High-School Textbooks, *in* 'Proceedings of the Annual Meeting of the Cognitive Science Society'.

McDonald, D. (2015), 'corpkit: a toolkit for corpus linguistics'.

McEnery, T., Baker, J. P. & Wilson, A. (1995), 'A statistical analysis of corpus based computer vs traditional human teaching methods of part of speech analysis', *Computer Assisted Language Learning* **8**(2-3), 259–274.

Mckee, G., Malvern, D. & Richards, B. (2000), 'Measuring Vocabulary Diversity Using Dedicated Software', *Literary and Linguistic Computing* **15**(3), 323–338.

McNamara, D. S., Cai, Z. & Louwerse, M. M. (2007), 'Optimizing LSA measures of cohesion', *Handbook of Latent Semantic Analysis* pp. 379–400.

McNamara, D. S., Crossley, S. A. & McCarthy, P. M. (2010), 'Linguistic features of writing quality', *Written Communication* **27**(1), 57–86.

McNamara, D. S., Graesser, A. C., McCarthy, P. M. & Cai, Z. (2014), *Automated Evaluation of Text and Discourse with Coh-Metrix*, Cambridge University Press, New York.

Měchura, M. (2017), Introducing Lexonomy: an open-source dictionary writing and publishing system, *in* 'Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference', pp. 19–21.

Michalke, M. (2018), *koRpus: An R Package for Text Analysis*. (Version 0.11-5).

Mikolov, T., Corrado, G., Chen, K. & Dean, J. (2013), 'Efficient Estimation of Word Representations in Vector Space', *Proceedings of the International Conference on Learning Representations (ICLR 2013)* pp. 1–12.

Miller, T. (2003), 'Essay assessment with Latent Semantic Analysis', *Journal of Educational Computing Research* **29**(4), 495–512.

Mohammad, S. M., Salameh, M. & Kiritchenko, S. (2016), 'How translation alters sentiment', *Journal of Artificial Intelligence Research* **55**, 95–130.

Montemurro, M. a. (2001), 'Beyond the Zipf-Mandelbrot law in quantitative linguistics', *Physica A: Statistical Mechanics and its Applications* **300**(3-4), 567–578.

Müller, S. (2004), '"Well you know that type of person": Functions of well in the speech of American and German students', *Journal of Pragmatics* **36**(6), 1157–1182.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0378216604000414*

Müller, T., Cotterell, R., Fraser, A. & Schütze, H. (2015), Joint lemmatization and morphological tagging with lemming, *in* 'Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing', pp. 2268–2274.

Myers, S. A. (2003), 'Reassessing the "proofreading trap": ESL tutoring and writing instruction', *The Writing Center Journal* **24**(1).

Nakamaru, S. (2010), 'Lexical issues in writing center tutorials with international and us-educated multilingual writers', *Journal of Second Language Writing* **19**(2), 95–113.

Nardello, M. (2016), *WordStatix Manual 1.9.0. (English*.
URL: *https://github.com/murat-aka/wordstatix/blob/master/Manuals/manual-wordstatix-en_1.9.0.odt*

Navarro, G. (2001), 'A guided tour to approximate string matching', *ACM computing surveys (CSUR)* **33**(1), 31–88.

Nerbonne, J., Dokter, D. & Smit, P. (1998), 'Morphological processing and computer-assisted language learning', *Computer Assisted Language Learning* **11**(5), 543–559.

Nesselhauf, N. (2003), 'The use of collocations by advanced learners of English and some implications for teaching', *Applied Linguistics* **24**(2), 223–242.

Neubauer-Petzoldt, R. (2016), Modelle der schreibprozessforschung und ihre relevanz für die schreibberatung und schreibpraxis in den natur-und ingenieurwissenschaften, *in* 'Wissenschaftliches Schreiben in Natur-und Technikwissenschaften', Springer, pp. 85–106.

Newman, M. E. J. (2004), 'Power laws, Pareto distributions and Zipf's law', *arXiv* (1), 28.

Oakes, M. P. (1998), *Statistics for Corpus Linguistics*, Edinburgh University Press, Edinburgh.

Palmquist, M. (2019), 'Directions in writing analytics: Some suggestions', *Journal of Writing Analytics* **3**, 1–12.

Paquot, M. & Bestgen, Y. (2009), Distinctive words in academic writing: A comparison of three statistical tests for keyword extraction, *in* 'Corpora: Pragmatics and Discourse', Brill Rodopi, pp. 247–269.

Paquot, M. & Granger, S. (2012), 'Formulaic Language in Learner Corpora', *Annual Review of Applied Linguistics* **32**, 130–149.

Pareto, V. (1964), *Cours d'économie politique*, Vol. 1, Librairie Droz.

Pearce, D. (2001), Synonymy in collocation extraction, *in* 'Proceedings of the workshop on WordNet and other lexical resources, second meeting of the north american chapter of the association for computational linguistics', pp. 41–46.

Pütz, T., De Kok, D., Pütz, S. & Hinrichs, E. (2018), Seq2seq or perceptrons for robust lemmatization. an empirical examination, *in* 'Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), December 13–14, 2018, Oslo University, Norway', number 155, Linköping University Electronic Press, pp. 193–207.

Pydde, M. & Girgensohn, K. (2011), 'Schreibberatung durch Peer Tutorinnen: Herausforderungen in Theorie und Praxis', *Textwissen und Schreibbewusstsein: Beiträge aus Forschung und Praxis* **6**, 263.

R Core Team (2021), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Ramineni, C. & Williamson, D. (2018), 'Understanding Mean Score Differences Between the e-rater Automated Scoring Engine and Humans for Demographically Based Groups in the GRE General Test', **ETS RR–18-12**(April).

Raymond, M. & Rousset, F. (1995), 'An exact test for population differentiation', *Evolution* **49**(6), 1280–1283.

Razali, N. M., Wah, Y. B. et al. (2011), 'Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests', *Journal of statistical modeling and analytics* **2**(1), 21–33.

Römer, U. (2006), 'Pedagogical applications of corpora: Some reflections on the current scope and a wish list for future developments', *Zeitschrift für Anglistik und Amerikanistik* **2**, 121–134.

Römer, U. & Arbor, A. (2009), 'English in academia: Does nativeness matter', *Anglistik: International Journal of English Studies* **2**(September), 89–100.

Ruhmann, G. (1995), Schreibprobleme-Schreibberatung, *in* 'Schreiben', Springer, pp. 85–106.

Russell, S. J. & Norvig, P. (2010), *Artificial intelligence*, 3 edn, Pearson.

Rychlỳ, P. (2008), A lexicographer-friendly association score., *in* 'RASLAN', pp. 6–9.

Rychlỳ, P. & Kilgarriff, A. (2007), An efficient algorithm for building a distributional thesaurus (and other sketch engine developments), *in* 'Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions', pp. 41–44.

Sachverständigenrat zur Begutachtung der Gesamtwirtschaftlichen Entwicklung (2009), *Die Zukunft nicht aufs Spiel setzen - Jahresgutachten 2009/2010*, Kohlhammer., Stuttgart.

Savickỳ, P. & Hlavácová, J. (2002), 'Measures of word commonness', *Journal of Quantitative Linguistics* **9**(3), 215–231.

Schiller, A., Teufel, S., Thielen, C. & Stöckert, C. (1999), Guidelines für das tagging deutscher textcorpora mit stts, Technical report, Technical Report. Institut für maschinelle Sprachverarbeitung, Stuttgart.

Schmid, H. (1994), 'Probabilistic Part-of-Speech Tagging Using Decision Trees'.

Schmid, H. (1999), Improvements in Part-of-Speech tagging with an application to German, *in* 'Natural Language Processing using very large Corpora', Springer, pp. 13–25.

Schmid, H. (2013), Probabilistic Part-of-Speech tagging using Decision Trees, *in* 'New Methods in Language Processing', p. 154.

Schroth-Wiechert, S. (2011), *Deutsch als Fremdsprache in den Ingenieurwissenschaften - Formulierungshilfen für schriftliche Arbeiten in Studium und Beruf*, Cornelsen Verlag, Berlin.

Scott, M. (1998/2019), *WordSmith tools manual*, Lexical Analysis Software.

Sha, G. (2010), 'Using Google as a super corpus to drive written language learning: a comparison with the British National Corpus', *Computer Assisted Language Learning* **23**(5), 377–393.

Shang, H. & Merrettal, T. (1996), 'Tries for approximate string matching', *IEEE Transactions on Knowledge and Data Engineering* **8**(4), 540–547.

Shao, C., Feng, Y. & Chen, X. (2018), 'Greedy search with probabilistic n-gram matching for neural machine translation', *arXiv* .

Shapiro, S. S. & Wilk, M. B. (1965), 'An analysis of variance test for normality (complete samples)', *Biometrika* **52**(3/4), 591–611.

Sheeba, J., Vivekanandan, K., Sabitha, G. & Padmavathi, P. (2013), Unsupervised hidden topic framework for extracting keywords (synonym, homonym, hyponymy and polysemy) and topics in meeting transcripts, *in* 'Advances in Computing and Information Technology', Springer, pp. 299–307.

Shermis, M. D. & Burstein, J. (2013), *Handbook of automated essay evaluation: Current applications and new directions*, Routledge.

Sheskin, D. J. (2003), *Handbook of Parametric and Nonparametric Statistical Procedures: Third Edition*, CRC Press.

Siyanova, A. & Schmitt, N. (2008), 'L2 Learner Production and Processing of Collocation: A Multi-study Perspective', *Canadian Modern Language Review/ La Revue canadienne des langues vivantes* **64**(3), 429–458.

Socher, R., Manning, C. D. & Ng, A. Y. (2010), Learning continuous phrase representations and syntactic parsing with recursive neural networks, *in* 'Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop', Vol. 2010, pp. 1–9.

Somasundaran, S., Burstein, J. & Chodorow, M. (2014), Lexical chaining for measuring discourse coherence quality in test-taker essays, *in* 'Proceedings of COLING 2014, the 25th International conference on computational linguistics: Technical papers', pp. 950–961.

Spaulding, J. & Morris, K. (2021), *rgeoprofile: Geographic Profiling Methods for Serial Crime Analysis*. R package version 0.2.2.

Spielmann, D. (2011), 'Schreibprobleme internationaler Studierender in der Schreibberatung', *Textwissen und Schreibbewusstsein, Beiträge aus Forschung und Praxis* pp. 317–334.

Srikant, R. & Agrawal, R. (1995), Mining generalized association rules, *in* 'Proceedings of the 21st VLDB Conference', IBM Research Division.

Stamatatos, E. (2009), 'A Survey of Modern Authorship Attribution Methods', *Journal of the American Society for Information Science and Technology* **60**(3), 538–556.

Steinhart, D. J. (2001), *Summary Street: An intelligent tutoring system for improving student writing through the use of Latent Semantic Analysis*, University of Colorado at Boulder.

Steinhoff, T. (2010), *Wissenschaftliche Textkompetenz - Sprachgebrauch und Schreibentwicklung in wissenschaftlichen Texten von Studenten und Experten*, Walter de Gruyter, Berlin.

Stevens, K., Kegelmeyer, P., Andrzejewski, D. & Buttler, D. (2012), Exploring topic coherence over many models and many topics, *in* 'Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning', Association for Computational Linguistics, pp. 952–961.

Sturges, H. A. (1926), 'The choice of a class interval', *Journal of the American Statistical Association* **21**(153), 65–66.

Sukkarieh, J. Z., Pulman, S. G. & Raikes, N. (2003), 'Automarking: using computational linguistics to score short ,free- text responses'.

The Qt Company (n.d.), 'Qt'.
URL: *https://www.qt.io*

Trapmann, S., Hell, B., Weigand, S. & Schuler, H. (2007), 'Die validität von schulnoten zur vorhersage des studienerfolgs-eine metaanalyse', *Zeitschrift für pädagogische Psychologie* **21**(1), 11–27.

Tsai, F. S. (2011), 'Text mining and visualisation of Protein-Protein Interactions.', *International journal of computational biology and drug design* **4**, 239–244.

Tschichold, C. et al. (2003), 'Lexically driven error detection and correction', *Calico Journal* **20**(3), 549.

Turney, P. D. (2001), Mining the web for synonyms: Pmi-ir versus lsa on toefl, *in* 'European conference on machine learning', Springer, pp. 491–502.

Ulmi, M., Bürki, G., Marti, M. & Verhein-Jarren, A. (2017), *Textdiagnose und Schreibberatung: Fach-und Qualifizierungsarbeiten begleiten*, Vol. 8544, UTB.

Ulmi, M., Bürki, G., Marti, M. & Verhein-Jarren, A. (2017), *Textdiagnose und Schreibberatung - Fach- und Qualifizierungsarbeiten begleiten*, 2. aufl. edn, UTB, Parderborn, München.

Van der Loo, M. P. (2014), 'The stringdist package for approximate string matching', *The R Journal* **6**(1), 111–122.

Venables, W. & Ripley, B. (2002), *Modern applied statistics with S*, number March, Springer.

Venohr, E. & Neis, C. (2013), 'Die Textsorte Vorlesungsprotokoll und ihre Relevanz für das wissenschaftliche Schreiben in der Fremdsprache Deutsch', *Informationen Deutsch als Fremdsprache* **40**(1).

Verity, R., Stevenson, M. D., Rossmo, D. K., Nichols, R. A. & Comber, S. C. L. (2014), 'Spatial targeting of infectious disease control: identifying multiple, unknown sources'.

Verspoor, M., Lowie, W., Dijk, M. V. & Van Dijk, M. (2008), 'Variability in Second Language Development From a Dynamic Systems Perspective', *The Modern Language Journal* **92**, 214–231.

Vitartas, P., Heath, J., Midford, S., Ong, K.-L., Alahakoon, D. & Sullivan-Mort, G. (2016), Applications of Automatic Writing Evaluation to Guide the Understanding of Learning and Teaching, *in* '33rd International Conference of Innovation, Practice and Research in the Use of Educational Technologies in Tertiary Education', p. 592.

Vyatkina, N. (2012), 'The Development of Second Language Writing Complexity in Groups and Individuals: A Longitudinal Learner Corpus Study', *The Modern Language Journal* **96**(4), 576–598.

Wang, R., Utiyama, M., Goto, I., Sumita, E., Zhao, H. & Lu, B.-L. (2016), 'Converting continuous-space language models into n-gram language models with efficient bilingual pruning for statistical machine translation', *ACM Transactions on Asian and Low-Resource Language Information Processing* **15**(3), 11.

Wang, T. & Hirst, G. (2010), Near-synonym lexical choice in latent semantic space, *in* 'Proceedings of the 23rd International Conference on Computational Linguistics', Association for Computational Linguistics, pp. 1182–1190.

Wang, X. & Ren, P.-h. (2009), 'A Corpus-based Differentiation of English Synonyms [J]', *Journal of Henan Polytechnic University (Social Sciences)* **1**.

Welbers, K. & van Atteveldt, W. (2020), *corpustools: Managing, Querying and Analyzing Tokenized Text*. R package version 0.4.2.

Westerlund, M. (2019), 'Correlations Between Textual Features and Grades on the Swedish National Exam in English: A Coh-Metrix Analysis'.

Wickham, H. (2019), *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0.

Wild, F. (2015), *lsa: Latent Semantic Analysis*. R package version 0.73.1.

Winkelmann, R. (2008), *Econometric Analysis of Count Data*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Witten, I. H., Frank, E. & Hall, M. a. (2011), *Data Mining*, Morgan Kaufmann.

Wolfe, C. R., Widmer, C. L., Torrese, C. V. & Dandignac, M. (2018), 'A Method for Automatically Analyzing Intelligent Tutoring System Dialogues with Coh-Metrix', *Journal of Learning Analytics* **5**(3), 222–234.

Wooldridge, J. (2010), *Econometric Analysis of Cross Section and Panel Data*, MIT Press.

Wooldridge, J. M. (2002), *Introductory Ecometrics*, South-Western.

Yoshimura, Y., Sobolevsky, S., Bautista Hobin, J. N., Ratti, C. & Blat, J. (2018), 'Urban association rules: uncovering linked trips for shopping behavior', *Environment and Planning B: Urban Analytics and City Science* **45**(2), 367–385.

Yuan, X. (2017), An improved apriori algorithm for mining association rules, *in* 'AIP conference proceedings', Vol. 1820, AIP Publishing.

Zamora-Martinez, F. & Castro-Bleda, M. J. (2018), 'Efficient Embedded Decoding of Neural Network Language Models in a Machine Translation System', *International Journal of Neural Systems* **28**(09), 1850007.

Zar, J. H. (1987), 'A fast and efficient algorithm for the fisher exact test', *Behavior Research Methods, Instruments, & Computers* **19**(4), 413–414.

Zegenhagen, J. (2008), 'Schreibberatung als spezielle form der lernberatung. auf dem weg zum 'autonomen autor'', *Autonomes Fremdsprachenlernen in Hochschule und Erwachsenenbildung: Erträge des 1. Bremer Symposions zum autonomen Fremdsprachenlernen (Fremdsprachen in Lehre und Forschung)* **1**, 156–168.

Zhang, J. & Liu, P. (2005), 'Corpus-based approaches to the differentiation of english synonyms', *Journal of PLA University of Foreign Languages* **28**(6).

Zhao, Y. (2013), 'R and Data Mining: Examples and Case Studies - RDataMining.com: R and Data Mining', (December 2012), 256.

Zipf, G. K. (1949), 'Human behavior and the principle of least effort.'.

Anhang

Zielgruppenanalyse

Tabelle 6.1: Erwartete Anzahl internationaler Studierender pro Campus und Fakultät

Campus	FArc	FBau	FElt	FMas	FMat	FNat	FPhi	FWir
11	0,888	26,481	145,404	151,617	33,778	19,588	30,868	7,926
12	0	0,212	22,180	2,213	1,621	0	34,048	0,917
15	0,222	0,741	4,929	6,917	1,081	1,518	38,579	76,310
16	0	0	0	0	0	0,228	0,801	0
17	0	0	0	0	0	0,456	0	0
18	0	0	0	0	0,270	0,911	16,498	0
25	0,222	0	0	1,107	0,090	33,406	0,183	0,066
27	0	0,106	2,464	1,107	0,180	13,210	0,137	0
31	3,994	26,270	4,929	13,280	0,180	9,566	17,024	0,852
32	2,219	0,106	0	10,237	0	0,531	0	0
34	1,775	87,389	152,798	101,263	0,991	3,720	9,656	1,769
37	0	0,212	118,295	9,684	6,936	0,683	0	0,852
41	31,510	0,106	0	1,107	3,873	49,729	0,206	0,131
42	36,170	0,636	0	1,383	0	6,074	0	0,066
63	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0
81	0	0,741	0	76,085	0	0,380	0	1,114
89	0	0	0	0	0	0	0	0

Tabelle 6.2: Erwartete CP aus schriftlichen Leistungen pro Campus und Fakultät

Campus	FArc	FBau	FElt	FMas	FMat	FNat	FPhi	FWir
11	0,35	5,95	20,11	23,78	5,40	4,21	25,23	1,06
12	0	0,05	3,07	0,35	0,26	0	27,83	0,12
15	0,09	0,17	0,68	1,08	0,17	0,33	31,53	10,17
16	0	0	0	0	0	0,05	0,65	0
17	0	0	0	0	0	0,10	0	0
18	0	0	0	0	0,04	0,20	13,48	0
25	0,09	0	0	0,17	0,01	7,18	0,15	0,01
27	0	0,02	0,34	0,17	0,03	2,84	0,11	0
31	1,57	5,90	0,68	2,08	0,03	2,06	13,91	0,11
32	0,87	0,02	0	1,61	0	0,11	0	0
34	0,70	19,63	21,14	15,88	0,16	0,80	7,89	0,24
37	0	0,05	16,36	1,52	1,11	0,15	0	0,11
41	12,37	0,02	0	0,17	0,62	10,69	0,17	0,02
42	14,20	0,14	0	0,22	0	1,31	0	0,01
63	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0
81	0	0,17	0	11,93	0	0,08	0	0,15
89	0	0	0	0	0	0	0	0

Tabelle 6.3: Distanz zwischen Hochschulstandorten in Meter

	11	12	15	16	17	18	25	27	31	32	34	37	41	42	63	73	81	89
11	-	466	612	687	430	520	403	446	495	1169	734	693	1450	1932	4626	16877	8383	8383
12	-	-	198	252	217	611	855	715	954	1636	1184	1110	1917	2394	4160	16440	8842	8842
15	-	-	-	79	216	587	1013	909	1074	1767	1344	1286	2046	2505	4046	16269	8943	8943
16	-	-	-	-	295	656	1088	967	1153	1846	1419	1355	2124	2584	3966	16197	9022	9022
17	-	-	-	-	-	401	832	796	866	1562	1161	1121	1838	2292	4260	16451	8727	8727
18	-	-	-	-	-	-	821	966	717	1382	1105	1133	1641	2041	4556	16594	8432	8432
25	-	-	-	-	-	-	-	397	285	810	331	313	1090	1597	4997	17281	8041	8041
27	-	-	-	-	-	-	-	-	670	1158	623	475	1428	1951	4727	17126	8370	8370
31	-	-	-	-	-	-	-	-	-	696	416	515	973	1439	5113	17290	7889	7889
32	-	-	-	-	-	-	-	-	-	-	539	719	281	793	5795	17975	7231	7231
34	-	-	-	-	-	-	-	-	-	-	-	191	805	1330	5314	17611	7750	7750
37	-	-	-	-	-	-	-	-	-	-	-	-	973	1502	5197	17551	7900	7900
41	-	-	-	-	-	-	-	-	-	-	-	-	-	530	6076	18234	6952	6952
42	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6550	18589	6451	6451
63	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12743	12985	12985
73	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24489	24489
81	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Korpora

Terminology Extraction

```
typicalTerms = data.frame()
for(i in 1:length(mergedWordList[,1])){
    typicalTerms[i,1] = mergedWordList[i,1]
    for(ii in 1:8){
        temp = data.frame()
        temp[1,1] = corpusSize[9]
        temp[2,1] = sum(mergedWordList[i,2:9])
        temp[1,2] = corpusSize[ii]
        temp[2,2] = mergedWordList[i,ii+1]
        typicalTerms[i,ii*3-1] = chisq.test(temp)[1]
        typicalTerms[i,ii*3] = chisq.test(temp)[3]
        typicalTerms[i,ii*3+1] = (temp[2,2]/temp[1,2])-(temp[2,1]/temp
            [1,1])
    }
}
```

Dissertationen der Fakultät für Architektur

Tabelle 6.4: Signifikant häufiger, bzw. seltener an der FArc benutzte Verben und Adjektive jeweils mit χ^2- und p-Wert

Verb	χ^2-Wert	p-Wert	Differenz	Adjektiv	χ^2-Wert	p-Wert	Differenz
koennen	147,55	0	0,00030	neue	734,83	0	0,00026
muessen	169,48	0	0,00014	Moderne	3.821,59	0	0,00023
entwickelt	80,01	0	0,00008	neuen	539,42	0	0,00022
entwickeln	180,60	0	0,00008	oeffentlichen	1.358,37	0	0,00019
entstehen	108,42	0	0,00007	raeumlichen	1.304,21	0	0,00018
vorgesehen	305,90	0	0,00006	raeumliche	917,20	0	0,00016
gelingt	227,98	0	0,00006	neu	379,98	0	0,00013
schaffen	240,54	0	0,00006	ggf	948,94	0	0,00012
gegenueber	21,55	0	0,00006	aktuellen	407,45	0	0,00011
einbezogen	221,22	0	0,00006	anderen	37,31	0	0,00010
...	...	...	...	...	...	...	
erkennen	132,01	0	-0,00012	mittleren	63,17	0	-0,00006
berechnet	167,43	0	-0,00012	optischen	102,56	0	-0,00006
erfolgte	94,34	0	-0,00012	hohen	40,92	0	-0,00007
ergibt	127,26	0	-0,00013	entfernt	91,65	0	-0,00007
fuehrt	110,81	0	-0,00013	direkt	59,68	0	-0,00008
gezeigt	152,38	0	-0,00015	gemessenen	137,22	0	-0,00009
zeigt	94,38	0	-0,00015	Anschliessend	159,16	0	-0,00011
siehe	130,92	0	-0,00016	anschliessend	171,65	0	-0,00013
zeigen	184,30	0	-0,00018	verwendeten	173,84	0	-0,00014
verwendet	201,24	0	-0,00021	waehrend	269,07	0	-0,00028

Tabelle 6.5: Signifikant häufiger, bzw. seltener an der FArc benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
fuer	614,09	0	0,00131	und	2.529,02	0	0,00535
auch	306,17	0	0,00068	oder	1.499,75	0	0,00123
aber	337,42	0	0,00038	&	1.045,03	0	0,00039
nur	68,93	0	0,00023	sowie	223,15	0	0,00033
immer	255,18	0	0,00021	wie	82,97	0	0,00027
mehr	124,24	0	0,00018	sondern	47,10	0	0,00010
insbesondere	271,49	0	0,00017	wenn	38,94	0	0,00010
d.h.	236,27	0	0,00012	ob	14,97	0,00011	0,00005
besonders	100,47	0	0,00010	weil	8,06	0,00452	0,00003
moeglichst	148,54	0	0,00009	sofern	39,66	0	0,00002
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
oben	25,88	0	-0,00005	Sowohl	6,26	0,01234	-0,00001
ebenfalls	40,86	0	-0,00009	wohingegen	12,58	0,00039	-0,00001
also	37,27	0	-0,00009	bevor	10,04	0,00153	-0,00001
nun	94,82	0	-0,00010	Jedoch	14,77	0,00012	-0,00002
zunaechst	69,74	0	-0,00010	Nachdem	20,20	0,00001	-0,00002
so	15,71	0	-0,00010	beziehungsweise	44,40	0	-0,00003
etwa	62,03	0	-0,00011	sowohl	14,37	0,00015	-0,00005
somit	95,02	0	-0,00013	Da	38,43	0	-0,00010
dann	69,98	0	-0,00014	da	49,95	0	-0,00014
jedoch	77,45	0	-0,00017	dass	150,08	0	-0,00050

Tabelle 6.6: Signifikant häufiger, bzw. seltener an der FBau benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
fuer	523, 39	0	0, 00084	sowie	43, 37	0	0, 00010
sehr	62, 12	0	0, 00010	da	19, 13	0	0, 00006
jedoch	29, 56	0	0, 00007	Da	14, 12	0, 00017	0, 00004
beispielsweise	64, 13	0	0, 00005	dass	0, 68	0, 40992	0, 00002
d.h.	82, 15	0	0, 00005	Sofern	132, 84	0	0, 00002
insbesondere	39, 07	0	0, 00005	beziehungsweise	29, 80	0	0, 00002
Weiterhin	61, 84	0	0, 00004	sofern	26, 54	0	0, 00001
etc.	64, 57	0	0, 00003	Sowohl	3, 11	0, 07759	0, 00001
somit	7, 53	0, 00607	0, 00003	Falls	6, 69	0, 00972	0, 00001
gleich	25, 54	0	0, 00003	Sobald	5, 76	0, 01642	0, 000004
...	...	...	...	...	...	...	
doch	166, 08	0	-0, 00006	Dass	75, 97	0	-0, 00003
hier	27, 19	0	-0, 00007	Aber	97, 51	0	-0, 00004
So	60, 89	0	-0, 00007	weil	50, 67	0	-0, 00005
immer	63, 24	0	-0, 00007	indem	73, 75	0	-0, 00005
so	16, 84	0	-0, 00007	wenn	24, 23	0	-0, 00006
hin	109, 10	0	-0, 00007	Und	175, 71	0	-0, 00006
selbst	221, 11	0	-0, 00012	denn	255, 51	0	-0, 00009
noch	98, 84	0	-0, 00014	sondern	153, 41	0	-0, 00013
also	205, 04	0	-0, 00015	oder	39, 39	0	-0, 00014
aber	360, 70	0	-0, 00027	und	4, 78	0, 02878	-0, 00016

Tabelle 6.7: Signifikant häufiger, bzw. seltener an der FElt benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
fuer	783,95	0	0,00140	dass	397,78	0	0,00078
jedoch	879,99	0	0,00055	wie	208,01	0	0,00041
so	279,93	0	0,00040	oder	118,68	0	0,00033
nur	207,46	0	0,00038	da	236,08	0	0,00029
somit	438,73	0	0,00028	Da	378,84	0	0,00029
sehr	136,32	0	0,00021	wenn	295,88	0	0,00027
bereits	129,39	0	0,00017	ob	75,44	0	0,00010
auch	21,26	0	0,00017	falls	187,37	0	0,00004
beispielsweise	244,51	0	0,00014	indem	27,12	0	0,00004
nun	202,08	0	0,00014	entweder	34,10	0	0,00004
...	...	...	...	...	...	...	
zwar	12,93	0,00032	-0,00003	und/oder	24,90	0	-0,00002
Auch	7,70	0,00552	-0,00004	Denn	22,90	0	-0,00002
eher	25,17	0	-0,00004	Doch	39,54	0	-0,00003
schon	13,75	0,00021	-0,00004	Aber	46,18	0	-0,00004
jetzt	50,16	0	-0,00004	Und	129,50	0	-0,00007
mal	78,15	0	-0,00005	denn	87,10	0	-0,00007
ganz	60,47	0	-0,00005	weil	65,20	0	-0,00007
ja	92,14	0	-0,00006	sondern	27,65	0	-0,00008
doch	92,40	0	-0,00006	&	99,48	0	-0,00011
So	31,05	0	-0,00007	sowie	114,46	0	-0,00022

Tabelle 6.8: Signifikant häufiger, bzw. seltener an der FMas benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
fuer	736,19	0	0,00091	sowie	276,16	0	0,00023
somit	387,70	0	0,00018	da	89,19	0	0,00012
jedoch	172,39	0	0,00016	Da	87,07	0	0,00009
so	95,51	0	0,00016	sowohl	41,69	0	0,00005
sehr	172,20	0	0,00016	Sowohl	3,34	0,06763	0,00001
insbesondere	389,43	0	0,00013	Sobald	5,24	0,02213	0,000003
links	663,49	0	0,00011	sobald	2,94	0,08658	0,000003
rechts	682,67	0	0,00011	respektive	6,29	0,01216	0,000003
hier	66,48	0	0,00010	Weil	1,13	0,28793	0,000002
nahezu	376,26	0	0,00009	solange	0,64	0,42369	0,000002
...	...	...	...	...	...	...	...
ganz	195,98	0	-0,00006	Aber	141,44	0	-0,00004
doch	258,27	0	-0,00007	wie	14,20	0,00016	-0,00007
immer	112,51	0	-0,00009	Und	336,22	0	-0,00007
dann	102,77	0	-0,00011	denn	288,05	0	-0,00009
mehr	143,78	0	-0,00012	wenn	71,55	0	-0,00009
selbst	312,47	0	-0,00013	weil	277,47	0	-0,00010
also	192,83	0	-0,00013	ob	203,22	0	-0,00010
noch	136,70	0	-0,00015	sondern	191,29	0	-0,00013
aber	499,63	0	-0,00029	&	437,07	0	-0,00015
auch	158,44	0	-0,00031	und	6,61	0,01013	-0,00017

Tabelle 6.9: Signifikant häufiger, bzw. seltener an der FMat benutzte Verben und Adjektive jeweils mit χ^2- und p-Wert

Verb	χ^2-Wert	p-Wert	Differenz	Adjektiv	χ^2-Wert	p-Wert	Differenz
ergibt	3.382,74	0	0,00043	optischen	8.232,34	0	0,00042
siehe	2.337,21	0	0,00042	optische	2.911,58	0	0,00017
gilt	3.043,08	0	0,00039	verwendeten	548,65	0	0,00015
F	2.803,95	0	0,00036	experimentellen	1.522,16	0	0,00015
erzeugt	2.497,18	0	0,00023	elektrischen	1.671,40	0	0,00015
entspricht	1.424,79	0	0,00023	gemessenen	931,17	0	0,00014
bestimmt	832,50	0	0,00023	kleiner	769,95	0	0,00012
laesst	604,02	0	0,00021	genau	697,26	0	0,00012
gezeigt	483,03	0	0,00017	linear	1.373,93	0	0,00011
betraegt	943,07	0	0,00016	experimentell	1.273,16	0	0,00011
...	...	...	...	...	...	...	...
getrocknet	425,33	0	-0,00008	eigene	337,46	0	-0,00007
versetzt	388,62	0	-0,00008	neue	186,89	0	-0,00008
koennte	167,21	0	-0,00008	anschliessend	189,04	0	-0,00008
fuehrte	374,72	0	-0,00009	organischen	411,70	0	-0,00008
nachgewiesen	300,23	0	-0,00009	signifikant	348,48	0	-0,00009
gegenueber	187,90	0	-0,00010	deutschen	523,47	0	-0,00010
zeigten	405,48	0	-0,00010	andere	235,75	0	-0,00010
eingesetzt	220,87	0	-0,00010	sozialen	628,89	0	-0,00012
zeigte	368,26	0	-0,00010	eigenen	585,11	0	-0,00012
durchgefuehrt	158,41	0	-0,00011	anderen	162,84	0	-0,00013

Tabelle 6.10: Signifikant häufiger, bzw. seltener an der FMat benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
so	838	0	0,00045	dass	103,29	0	0,00025
also	1.637,93	0	0,00038	Da	443,04	0	0,00020
etwa	1.305,20	0	0,00030	da	49,84	0	0,00008
nun	1.693,95	0	0,00026	falls	919,46	0	0,00007
somit	638,10	0	0,00022	beziehungsweise	301,56	0	0,00005
hier	218,77	0	0,00017	Falls	157,23	0	0,00002
dann	151,10	0	0,00012	indem	8,65	0,00327	0,00001
Dann	757,04	0	0,00009	wenn	1,04	0,30719	0,00001
sodass	675,42	0	0,00009	respektive	48,93	0	0,00001
allerdings	123,02	0	0,00008	bevor	4,86	0,02748	0,00001
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
jedoch	49,13	0	-0,00008	Aber	200,44	0	-0,00005
eher	320,51	0	-0,00008	denn	185,97	0	-0,00007
mehr	78,41	0	-0,00009	Und	288,74	0	-0,00007
schon	200,13	0	-0,00009	wie	26,40	0	-0,00009
selbst	310,99	0	-0,00012	weil	337,26	0	-0,00010
nur	58,34	0	-0,00013	ob	374,12	0	-0,00013
Auch	319,35	0	-0,00014	sondern	363	0	-0,00017
noch	186,35	0	-0,00017	sowie	292,13	0	-0,00022
fuer	32,59	0	-0,00018	&	1.058,59	0	-0,00022
aber	531,96	0	-0,00028	oder	2.004,36	0	-0,00083

Tabelle 6.11: Signifikant häufiger, bzw. seltener an der FNat benutzte Verben und Adjektive jeweils mit χ^2- und p-Wert

Verb	χ^2-Wert	p-Wert	Differenz	Adjektiv	χ^2-Wert	p-Wert	Differenz
erfolgte	2.437,58	0	0,00020	anschliessend	1.295,58	0	0,00012
durchgefuehrt	867,09	0	0,00013	Anschliessend	1.117,41	0	0,00010
zeigten	1.275,70	0	0,00010	organischen	1.090,82	0	0,00008
nachgewiesen	1.072,78	0	0,00010	signifikant	782,79	0	0,00007
zeigte	1.009,43	0	0,00009	waehrend	204,08	0	0,00007
untersucht	516	0	0,00009	entfernt	675,49	0	0,00007
gezeigt	547,66	0	0,00009	verschiedenen	205,70	0	0,00006
beobachtet	814,81	0	0,00009	weitere	169,42	0	0,00005
versetzt	1.195,68	0	0,00008	untersuchten	280,23	0	0,00005
getrocknet	1.274,33	0	0,00008	erhaltenen	406,31	0	0,00004
...	...	...	...	...	...	...	
stellt	415,08	0	-0,00006	genau	485,06	0	-0,00004
machen	1.060,85	0	-0,00007	andere	287,53	0	-0,00005
lassen	460,31	0	-0,00007	eigene	1.080,14	0	-0,00006
geht	909,20	0	-0,00007	politischen	1.679,71	0	-0,00006
gibt	506,91	0	-0,00008	neue	701,66	0	-0,00007
laesst	462,96	0	-0,00008	neuen	783,61	0	-0,00007
muessen	888,53	0	-0,00009	anderen	287,28	0	-0,00008
gilt	904,70	0	-0,00009	deutschen	2.195,44	0	-0,00009
ergibt	958,10	0	-0,00010	eigenen	2.014,33	0	-0,00010
koennen	584,16	0	-0,00017	sozialen	2.718,76	0	-0,00010

Tabelle 6.12: Signifikant häufiger, bzw. seltener an der FNat benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
ebenfalls	177,06	0	0,00005	&	341,25	0	0,00007
jeweils	140,14	0	0,00005	sowie	10,12	0,00147	0,00002
je	108,54	0	0,00003	wohingegen	67,45	0	0,00001
dreimal	346,52	0	0,00003	entweder	15,21	0	0,00001
zweimal	237,18	0	0,00002	sowohl	4,30	0,03805	0,00001
vermutlich	94,12	0	0,00002	Sowohl	24,36	0	0,00001
moeglicherweise	75,73	0	0,00002	Nachdem	18,46	0	0,00001
mindestens	47,21	0	0,00001	bevor	10,99	0	0,000005
zusaetzlich	18,50	0	0,00001	Jedoch	8,52	0,00351	0,000004
zuvor	22,26	0	0,00001	Obwohl	1,82	0,17782	0,000002
...	...	...	...	...	...	...	...
mehr	988,66	0	-0,00015	Wenn	809,58	0	-0,00006
selbst	2.291,82	0	-0,00015	Und	1.867,22	0	-0,00007
hier	880,50	0	-0,00016	denn	1.472,94	0	-0,00008
dann	1.174,65	0	-0,00016	weil	1.951,51	0	-0,00011
also	1.707,60	0	-0,00017	da	522,54	0	-0,00013
noch	955,24	0	-0,00018	sondern	2.404,06	0	-0,00020
nur	517,59	0	-0,00018	oder	513,36	0	-0,00021
aber	1.616	0	-0,00024	wenn	2.645,94	0	-0,00023
fuer	248,09	0	-0,00024	wie	1.511,41	0	-0,00033
so	2.270,17	0	-0,00034	dass	1.841,27	0	-0,00051

Tabelle 6.13: Signifikant häufiger, bzw. seltener an der FPhi benutzte Verben und Adjektive jeweils mit χ^2- und p-Wert

Verb	χ^2-Wert	p-Wert	Differenz	Adjektiv	χ^2-Wert	p-Wert	Differenz
machen	3.167,79	0	0,00021	sozialen	7.450,59	0	0,00034
geht	2.359,69	0	0,00020	eigenen	5.351,86	0	0,00029
gibt	787,26	0	0,00014	deutschen	5.504,74	0	0,00027
gab	2.183,19	0	0,00013	anderen	1.121,27	0	0,00023
steht	1.008,88	0	0,00012	politischen	5.306,30	0	0,00022
waere	789,42	0	0,00011	andere	1.252,50	0	0,00017
wuerde	915,81	0	0,00011	eigene	2.884,43	0	0,00017
gemacht	1.134,80	0	0,00010	neuen	1.502,07	0	0,00015
macht	1.159,93	0	0,00009	neue	1.315,42	0	0,00015
verstehen	1.556,17	0	0,00009	deutsche	2.775,12	0	0,00012
...	...	...	...	...	...	...	...
bestimmt	1.106,71	0	-0,00016	moeglich	333,04	0	-0,00010
eingesetzt	1.615,41	0	-0,00017	weitere	414,18	0	-0,00010
dargestellt	1.291,02	0	-0,00018	untersuchten	959,17	0	-0,00011
erfolgte	1.731,21	0	-0,00019	hohen	797,50	0	-0,00012
gezeigt	1.801,17	0	-0,00019	Anschliessend	1.457,35	0	-0,00012
erfolgt	1.672,75	0	-0,00020	verschiedenen	574,68	0	-0,00013
siehe	1.449,60	0	-0,00020	hohe	952,03	0	-0,00013
untersucht	1.911,55	0	-0,00021	anschliessend	1.696,33	0	-0,00015
durchgefuehrt	3.127,07	0	-0,00030	waehrend	710,45	0	-0,00018
verwendet	2.986,55	0	-0,00030	verwendeten	2.055,53	0	-0,00018

Tabelle 6.14: Signifikant häufiger, bzw. seltener an der FPhi benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
auch	8.288,03	0	0,00147	und	9.582,28	0	0,00424
aber	6.896,88	0	0,00074	dass	2.531,52	0	0,00085
noch	4.175,33	0	0,00056	wie	3.766,31	0	0,00076
so	2.256,16	0	0,00050	oder	2.121,51	0	0,00060
selbst	6.695,88	0	0,00043	sondern	7.086	0	0,00057
mehr	2.739,80	0	0,00036	wenn	3.676,33	0	0,00042
nur	1.028,33	0	0,00036	weil	5.080,43	0	0,00031
immer	3.794,73	0	0,00035	denn	4.644,17	0	0,00026
dann	2.298,99	0	0,00034	Und	5.389,78	0	0,00024
schon	2.926,34	0	0,00026	Aber	2.835,90	0	0,00015
...	...	...	...	...	...	...	...
nahezu	398,48	0	-0,00005	Sofern	18,62	0	-0,000004
Weiterhin	574,47	0	-0,00006	Falls	12,65	0	-0,000004
zusaetzlich	423,49	0	-0,00007	bevor	7,08	0,00778	-0,000005
etwa	160,85	0	-0,00007	falls	21,53	0	-0,00001
je	316,24	0	-0,00007	entweder	11,29	0	-0,00001
sehr	100,22	0	-0,00007	wohingegen	112,09	0	-0,00001
ebenfalls	455,25	0	-0,00011	Sowohl	61,53	0	-0,00002
somit	749,90	0	-0,00015	&	36,57	0	-0,00003
jeweils	880,99	0	-0,00015	sowohl	72,61	0	-0,00004
fuer	336,45	0	-0,00038	Da	594,81	0	-0,00015

Tabelle 6.15: Signifikant häufiger, bzw. seltener an der FWir benutzte Verben und Adjektive jeweils mit χ^2- und p-Wert

Verb	χ^2-Wert	p-Wert	Differenz	Adjektiv	χ^2-Wert	p-Wert	Differenz
koennen	457,70	0	0,00047	logistischen	6.030,59	0	0,00026
muessen	351,07	0	0,00018	einzelnen	305,20	0	0,00020
stellt	255,35	0	0,00016	anderen	163,86	0	0,00019
lassen	189,65	0	0,00015	empirischen	1.047,52	0	0,00013
betrachteten	836,51	0	0,00014	jeweiligen	190,26	0	0,00012
erfolgt	112,09	0	0,00012	externen	1.029,75	0	0,00012
besteht	143,34	0	0,00012	deutschen	273,50	0	0,00011
stellen	261,12	0	0,00012	positiv	425,18	0	0,00011
bewertet	684,15	0	0,00010	neue	148,62	0	0,00010
ermoeglicht	177,56	0	0,00010	kurzfristigen	2.486,98	0	0,00010
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
zeigten	271,01	0	-0,00012	verschiedenen	40,80	0	-0,00008
untersucht	128,56	0	-0,00013	verwendeten	74,98	0	-0,00008
zeigte	258,44	0	-0,00013	Anschliessend	114,46	0	-0,00008
bestimmt	153,57	0	-0,00014	deutlich	40,42	0	-0,00009
beobachtet	292,74	0	-0,00015	organischen	217,21	0	-0,00009
gezeigt	252,81	0	-0,00017	gemessenen	198,10	0	-0,00009
siehe	224	0	-0,00018	entfernt	202,37	0	-0,00010
durchgefuehrt	245,39	0	-0,00020	ueber	15,12	0	-0,00010
erfolgte	368,47	0	-0,00021	anschliessend	201,07	0	-0,00013
verwendet	310,26	0	-0,00023	waehrend	313,30	0	-0,00027

Tabelle 6.16: Signifikant häufiger, bzw. seltener an der FWir benutzte Adverbien und Konjunktionen jeweils mit χ^2- und p-Wert

Adverb	χ^2-Wert	p-Wert	Differenz	Konjunktion	χ^2-Wert	p-Wert	Differenz
auch	345,16	0	0,00064	dass	347,25	0	0,00068
also	407,86	0	0,00027	oder	453,99	0	0,00060
fuer	17,12	0	0,00019	sowie	537,05	0	0,00045
nur	61,69	0	0,00019	wenn	676,43	0	0,00038
aber	83,95	0	0,00017	ob	364,41	0	0,00020
So	193,01	0	0,00016	da	74,92	0	0,00015
insbesondere	274,13	0	0,00016	sondern	125,11	0	0,00015
eher	445,74	0	0,00015	weil	305,92	0	0,00015
dann	92,19	0	0,00014	sowohl	62,34	0	0,00009
somit	122,50	0	0,00014	denn	86,52	0	0,00007
...	...	...	...	...	...	...	...
noch	7,48	0,00623	-0,00005	wohingegen	3,67	0,05538	-0,00001
so	6,15	0,01313	-0,00006	Und	1,81	0,17906	-0,00001
links	117,84	0	-0,00006	nachdem	12,55	0,	-0,00001
rechts	122,87	0	-0,00006	beziehungsweise	9,18	0,00245	-0,00001
jeweils	30,71	0	-0,00007	Obwohl	5,41	0,02005	-0,00001
wieder	37,34	0	-0,00007	Nachdem	8,75	0,00310	-0,00001
schon	54,33	0	-0,00007	bevor	9,31	0,00227	-0,00001
ebenfalls	41,15	0	-0,00008	Da	10,57	0,00115	-0,00004
sehr	45,65	0	-0,00011	&	28,05	0	-0,00006
etwa	364,62	0	-0,00023	und	24,01	0	-0,00046

Eidesstattliche Erklärung

Hiermit versichere ich,

Tobias Gärtner
geboren am 29. Oktober 1988 in Gehrden,

die hier vorliegende Arbeit selbst angefertigt und alle für die Arbeit verwendeten Quellen und Hilfsmittel vollständig angegeben zu haben. Außerdem ist diese Arbeit bisher noch nicht als Prüfungsarbeit verwendet worden.

Hannover, 28. Februar 2023