J.B. METZLER

00_Hirschmann_JBM_Titelei|primustype|JBM|Hr.Pöhlmann|Hirschmann|Korpuslinguistik|29.04.2019

Hagen Hirschmann

Korpuslinguistik

Eine Einführung

Mit Abbildungen und Grafiken

J. B. Metzler Verlag

Der Autor
Hagen Hirschmann ist Dozent am Institut für deutsche Sprache und Linguistik der HU Berlin – am Lehrstuhl für Korpuslinguistik und Morphologie.

ISBN 978-3-476-02643-9
ISBN 978-3-476-05493-7 (eBook)
https://doi.org/10.1007/978-3-476-05493-7

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

J. B. Metzler

Einbandgestaltung: Finken & Bumiller, Stuttgart (Foto: shutterstock.com)

J. B. Metzler ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature
Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Inhaltsverzeichnis

Vorwort – wem kann dieses Buch helfen und wie? .....

1 Theoretische Grundlagen 1

1.1 Das Wesen der Korpuslinguistik und ihre Stellung in der Linguistik 1
1.2 Kriterien für Korpora (Definitionen) 2
1.3 Einordnung des Datentyps ›Korpusdaten‹ sowie korpusbasierter Forschung 4
1.4 Anwendungsbereiche von Korpora 7
1.6 Ein Entscheidungsbaum zur Verwendung von Korpora 17

2 Praxisteil I: Erstellung und Aufbau von Korpora 19

2.1 Datenakquise: Gewinnung von Korpus-Primärdaten 19
2.2 Vom geschriebenen Text zum Korpus: Erstellung von Textkorpora 21
2.3 WebLicht: eine Online-Plattform zur automatischen Verarbeitung von Korpusdaten 73
2.4 Vom gesprochenen Text zum Korpus: Erstellung von Gesprächskorpora 74
2.5 Evaluation von Korpusannotationen 96
2.6 Daten über die Daten: Annotation von Metadaten 102

3 Praxisteil II: Suchinterfaces, Anfragesprachen und Anfragemöglichkeiten 105

3.1 Suchwerkzeuge für eigens erstellte Daten 106
3.2 Online-Suchinterfaces für große Standardkorpora 156
3.3 Evaluation von Korpussuchen 172

4 Praxisteil III: Statistische Auswertung von Korpusdaten 177

4.1 Vorbereitung von auszuwertenden Daten: Datenexportformate und Konversionsszenarien 178
4.2 Vorbereitende Überlegungen: Typen von Statistik 181
4.3 Frequenzauswertungen: Erstellung von Frequenzlisten 182
4.4 Studientypen: verschiedene Klassifikationsansätze 185
4.5 Methoden für die kontrastive Analyse mindestens zweier Varietäten 186
4.6 Methoden für die Analyse einer bestimmten Varietät 208
4.7 Korrelationen 218

5 Serviceteil 223

5.1 Internetlinks zu frei verfügbaren Korpusressourcen 223
5.2 Zitierte Literatur 229
5.3 Sachregister 235

Vorwort – wem kann dieses Buch helfen und wie?

Die Korpuslinguistik ist seit etwa dem Jahr 2000 von einem technischen Spezialgebiet zu der empirischen Standardmethode überhaupt avanciert, durch die in der Linguistik sprachgebrauchsbezogene und auch theoretische Fragestellungen behandelt werden. Ab Mitte der 2000er Jahre existieren auch die ersten ›modernen‹ Einführungsbücher in die Korpuslinguistik, u. a. für deutschsprachige Korpusressourcen, vgl. z. B. Lemnitzer/Zinsmeister (2015, Erstausgabe 2006), Scherer (2006) oder Perkuhn/Keibel/Kupietz (2012) für deutschsprachige Einführungen. Warum also ein weiteres Buch in der Reihe dieser bereits vorhandenen, guten Ressourcen? Den Anstoß für das vorliegende Buch – und dafür danke ich ihr recht herzlich – gab die Kollegin Nanna Fuhrhop, die betonte, es müsse zusätzlich zu den ihr bekannten Einführungen ein Lehrbuch mit Handlungsaufgaben geben, anhand derer man die korpuslinguistischen Inhalte durch angeleitete, authentische Analysen erwirbt. Dieses Buch versucht diesem Wunsch zu entsprechen: Das Ziel ist es, noch unerfahrene Korpusnutzerinnen und -nutzer in die Handlungsfelder korpuslinguistischer Arbeit einzuführen, und zwar durch die entsprechenden Analysetätigkeiten selbst. So bilden praktische Anleitungen innerhalb der Kapitel sowie Aufgaben zur Erstellung und Bearbeitung von Korpusdaten am Ende der Kapitel das Kernstück der Wissensvermittlung in diesem Buch. Genau aus diesem Grund rate ich jeder Person, die sich mithilfe dieses Bandes korpuslinguistische Fachkompetenz aneignen will, dringend zur Bearbeitung der praktischen Anweisungen. Ohne sie ist ein nachhaltiger Erwerb der vermittelten Kenntnisse, Fähigkeiten und Fertigkeiten nicht zu erwarten.

Wie in allen wissenschaftlichen Analysen mit Interpretationsgehalt sind die zu diesem Buch verfügbaren Lösungen diskutabel. Es geht hierbei selten um ›die korrekte Lösung‹, sondern meistens um eine plausible oder mögliche Lösung. Achten Sie bei der Einsicht von Lösungen darauf – eine alternative Lösung muss nicht unbedingt falsch sein.

Nutzerinnen und Nutzer des Buchs benötigen keinerlei technische Vorkenntnisse, wohl aber linguistische. Es wird vorausgesetzt, dass die jeweils relevante linguistische Terminologie und insgesamt die Standardkonzepte der Phonologie, Morphologie, Syntax und Textlinguistik bekannt sind.

Die Programme und Dateiformate, die über dieses Buch vermittelt werden, sind in aller Regel plattformübergreifend verfügbar oder als Onlineressourcen plattformunabhängig. Bezahlpflichtige Dienste und Programme werden prinzipiell ausgespart, wodurch ggf. Ressourcen, die in bestimmten Forschungskontexten als relevant erachtet werden, außen vor bleiben.

Die Verwendung einer Onlineressource birgt grundsätzlich das Risiko, dass die zur Fertigstellung des Buchs geprüfte Adresse, unter der die Ressource verfügbar ist, irgendwann geändert oder gelöscht wird. Dieses Ri-

siko lässt sich leider nur bedingt minimieren. Sicherheitshalber finden Sie unter der Webadresse https://bit.ly/2TS0yg0 eine möglichst häufig aktualisierte Liste der im Buch erwähnten Internetressourcen. Sollte eine genannte Adresse nicht erreichbar sein, verwenden Sie bitte eine Internetsuche nach der genannten Ressource oder konsultieren Sie das genannte Dokument.

Wie in jedem wissenschaftlichen Fachbereich kann man auch bei korpuslinguistischen Belangen beliebig tief in Sachverhalte einsteigen. In diesem Buch wird angestrebt, eine breite Übersicht über die aktuellen korpuslinguistischen Standardverfahren zu vermitteln und dabei möglichst viele relevante Aspekte anzureißen. Zur Vertiefung können die genannten Literaturhinweise dienen.

Dank gebührt in erster Linie allen Personen, deren Arbeit an Korpusressourcen der wissenschaftlichen Forschung frei zur Verfügung steht und deshalb in diesem Buch zur Nutzung weiterempfohlen werden konnte. Kolleginnen und Kollegen, die diese Publikation aus fachlichen und technischen Gründen erst möglich gemacht haben, sind Anke Lüdeling, Carolin Odebrecht, Felix Golcher, Thomas Krause und Ute Hechtfischer vom J. B. Metzler Verlag – 1000 Dank!

1 Theoretische Grundlagen

1.1 Das Wesen der Korpuslinguistik und ihre Stellung in der Linguistik
1.2 Kriterien für Korpora (Definitionen)
1.3 Einordnung des Datentyps ›Korpusdaten‹ sowie korpusbasierter Forschung
1.4 Anwendungsbereiche von Korpora
1.5 Welches ist das passende Korpus für meine persönliche Fragestellung? – Korpora und ihre ›Repräsentativität‹
1.6 Ein Entscheidungsbaum zur Verwendung von Korpora

In diesem Kapitel werden wesentliche Definitionen zur korpuslinguistischen Methode vorgestellt, das Wesen der Korpuslinguistik in Bezug auf andere empirische Methoden und auf verschiedene linguistische Teilgebiete beschrieben sowie Vorüberlegungen zur Verwendung von Korpora formuliert. Dieses Kapitel ist das einzige, das nicht handlungsorientiert ist und nicht mit praktischen Analyseaufgaben versehen ist. Dennoch sollten es Einsteiger aufmerksam lesen, um bei der Planung und Durchführung eigener Forschungsprojekte Missverständnisse und methodische Fehler zu vermeiden.

1.1 | Das Wesen der Korpuslinguistik und ihre Stellung in der Linguistik

Definition

Die **Korpuslinguistik** ist eine empirische Methode mit dem Ziel, linguistische Forschungsfragen zu bearbeiten. Eine empirische Methode zeichnet sich durch bestimmte Datentypen aus, die gezielt durch experimentelle oder nicht-experimentelle Datenerhebungen gewonnen werden.

Korpuslinguistik ist also nicht zu verwechseln mit einer theoretischen Ausrichtung wie der kognitiven Linguistik. Empirische Methoden wie die Korpuslinguistik können theorieübergreifend genutzt werden und können gerade der Theoriebildung dienen.

Zwar ist die korpuslinguistische Methodologie, die in diesem Buch vorgestellt wird, nicht zwingendermaßen an computergestützte Verarbeitungsprozesse gebunden; die Entwicklung der Korpuslinguistik ist aber maßgeblich parallel zur Entwicklung computerbasierter Speicher- und Verarbeitungskapazitäten erfolgt. Während in den 1990er Jahren die ersten Programme zur Sprachanalyse aufkamen, die ausschließlich von Computerlinguistinnen und Computerlinguisten verwendet werden konnten, sind gerade in den letzten zehn Jahren Anwendungen geschaf-

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_1

fen worden, die Nutzerinnen und Nutzer mit normalen Computerkenntnissen verwenden können. Viele davon sind webbasiert und benötigen somit nicht einmal Kenntnisse zur Installation auf einem lokalen Rechner. Im vergangenen Jahrzehnt ist die Korpuslinguistik von einem technischen Spezialgebiet zu einer Standardmethode avanciert, die in vielen Universitätscurricula fest verankert ist. Aus diesem Grund besteht heute ein umso größerer Bedarf an unkomplizierten Zugängen zu korpuslinguistischen Methoden.

1.2 | Kriterien für Korpora (Definitionen)

Definition

Ein **Korpus** ist eine Sammlung von Textdaten, also Sprache im Kontext, die dem Zweck der linguistischen Auswertung dient und eine quantitative Auswertung von (qualitativen) sprachlichen Merkmalen zulässt.
Primärdaten bezeichnen die ursprünglich gesammelten Textdaten (ohne jegliche zusätzliche Information; s. Kap. 2.1 für eine erweiterte Definition und Problematisierung).
Metadaten bezeichnen Informationen über diese Daten, z. B. den Autor, das Erstellungsjahr der Primärtexte, den Namen der Korpusersteller usw. Metadaten müssen sich nicht auf sämtliche Primärdaten im Korpus beziehen, sondern können einzelne Teile des Korpus umfassen.
Annotationen bezeichnen Interpretationen der Primärdaten in Form linguistischer Kategorien (wie ›Nomen‹, ›Subjekt‹ oder ›Nebensatz‹).

Der Begriff ›Korpus‹ lässt sich enger definieren, wenn man genauere Anforderungen an den im Korpus gespeicherten Datentyp stellt. Nach der aktuellen Standardauffassung müssen die Sprachdaten digital (in einem computerlesbaren Format) vorliegen. Auch wenn bestimmte Linguistinnen und Linguisten eine analog verfügbare Textsammlung als Korpus bezeichnen, wird die Relevanz der digitalen Verfügbarkeit unmittelbar deutlich, wenn man sich vor Augen führt, dass auf digital gespeicherte Daten zurückgehende Analysen leichter nachvollziehbar und replizierbar gemacht werden können. Eine digitale Weitergabe von Quellen sowie deren Analysen ist technisch einfacher zu bewerkstelligen als analoge Datenübermittlungen. In Kapitel 2 wird gezeigt, dass die verschiedenen Datenformen im digital vorliegenden Korpus – Primärdaten, verschiedene voneinander abhängige oder unabhängige Annotationen sowie Metadaten – strukturell getrennt gespeichert werden und in der Korpusauswertung (s. Kap. 3 und 4) beliebig aufeinander bezogen werden können. Bei großen Datenmengen ist dies mit analogen Sprachdaten nicht möglich.

Die Mindestanforderung an die Sprachdaten ist, dass es sich um textuelle Daten, also um Sprache im Kontext handelt. Somit ergibt eine Wortliste, z. B. eine Sammlung von Verben, die mit dem Präfix *ver-* beginnen, kein Korpus. Medial kann es sich um mündliche oder schriftliche oder

auch um gebärdete Sprache handeln. Drei Audiodateien, die Gespräche zwischen Individuen darstellen, können also als Korpus bezeichnet werden, eine beliebige Anzahl aus dem Internet herauskopierter Textdateien ebenso.

Es gibt über die genannte Mindestanforderung hinaus weitere, optionale Kriterien für den Korpusbegriff: Häufig bestehen Korpora nicht nur aus den bloßen Textquellen – *Primärdaten* –, sondern aus Informationen über diese Quellen – *Metadaten* – und aus linguistischen Informationen über die sprachlichen Strukturen – *Annotationen*.

Zur Beschaffenheit von Primärdaten: Einige Korpuslinguisten stellen an die gesammelten Textdaten den Anspruch der Natürlichkeit und Authentizität (vgl. zu diesem Problem zusammenfassend Mukherjee 2009, S. 21). Dies ist eng verknüpft mit dem Erhebungskontext der Daten: Wissen die Textverfasser bei der Texterstellung, dass es sich um eine Korpusdatenerhebung handelt (wie es z. B. bei der Erhebung von Lernerdaten in der Regel der Fall ist) oder fallen der Zeitpunkt der Texterstellung und der Erhebungszeitpunkt (wie z. B. bei der Erstellung von Internetkorpora) auseinander? Das erste Szenario bedingt gegebenenfalls eine Beeinflussung der Textproduzenten und somit eine Minderung der Authentizität der Sprachdaten im Korpus. Doch je spezifischer die Anforderungen an die zu analysierenden Texte sind, umso mehr kann es erforderlich sein, die Bedingungen bzw. Vorgaben bei der Textproduktion zu kontrollieren und somit experimentelle Bedingungen zu schaffen. Wichtig zu bedenken ist hierbei, dass nur vor dem Hintergrund eines bestimmten Forschungsziels entschieden werden kann, wie stark in die Sprachproduktion eingegriffen werden darf. Für die Definition des Korpusbegriffs erscheint es wenig sinnvoll, den Grad der Authentizität der gesammelten Textdaten per se einzuschränken.

Zweck von Korpora: Korpora dienen unterschiedlichen Verwendungszielen. In der linguistischen Forschung sollen sie vor allem eine empirische Grundlage bieten, anhand welcher man Hypothesen testen kann. Sie helfen aber auch Lexikographen bei der Erstellung sprachspezifischer Lexika oder können von Unterrichtenden genutzt werden, um authentische Belege für bestimmte sprachliche Kategorien oder Strukturen aufzufinden und im Unterricht zu verwenden. In Kapitel 1.4.1 bis 1.4.2 werden diese Anwendungsfelder genauer vorgestellt. Dass ein Korpus einen linguistischen Zweck erfüllen muss, kann als notwendiges Kriterium für Korpora erachtet werden. Dieser Ansicht nach kann dieselbe Textgrundlage durchaus ein Korpus oder auch kein Korpus sein; z. B. stellt eine digital gespeicherte Romansammlung, die zur Unterhaltung gelesen wird, kein Korpus dar. Dient dieselbe Textsammlung der Bearbeitung einer linguistischen Forschungsfrage, muss sie als Korpus bezeichnet werden.

Marginalie: "optionale Kriterien" Nicht kann sie setzen, sondern normal formatiert.

Marginalie: "Authentizität von Korpora"

marginalie: "Korpora sollten einem Forschungszweck dienen"

Definition

Zusammenfassung: Kriterien für Korpora

Notwendige Kriterien:
- Textdaten (Wörter und Sätze im sprachlichen Kontext häufig: digital vorliegend und verarbeitbar)
- Linguistische Nutzung

} *hinreichendes* Kriterium

Optionale Merkmale:
- Metadaten
- Annotationen

1.3 | Einordnung des Datentyps ›Korpusdaten‹ sowie korpusbasierter Forschung

Aus dem vorangegangenen Kapitel 1.2 geht hervor, dass es sich bei Korpusdaten um Textdaten eines beliebigen medialen Typs sowie um beliebige Informationen zu diesen Textdaten handeln kann. Somit unterscheiden sich Korpusdaten entschieden von anderen Forschungsdaten. In diesem Kapitel geht es um die Benennung und Einordnung dieser Unterschiede.

Sprachdaten, die für linguistische Analysen verwendet werden, lassen sich in mehrerlei Hinsicht klassifizieren. Betrachtet man die Quelle der Daten und somit die Art, wie die Daten gewonnen wurden, muss man mindestens zwischen den folgenden Kategorien unterscheiden.

Introspektiv gewonnene bzw. auf Introspektion beruhende Sprachdaten sind sprachliche Beispiele für bestimmte sprachliche Phänomene, Zusammenhänge oder Regeln, die von der Betrachterin oder dem Betrachter selbst erdacht wurden. Solche Daten können offensichtliche Fälle von Grammatikalität darlegen und sind wichtig, um abstrakte linguistische Beschreibungen an konkreten Beispielen leichter zugänglich zu machen. Vergleichen Sie (1) und (2).

(1) *wegen dieser Sache – dieser Sache wegen*
(2) *aufgrund dieser Sache – *dieser Sache aufgrund*

Die Vor- und Nachstellung der Adposition *wegen* in (1) ist grammatisch. Die Nachstellung der Adposition *aufgrund* in (2) ist ungrammatisch.

In weniger offensichtlichen sprachlichen Zusammenhängen eignen sich solche Daten nicht, weil sie nicht objektiv sind und die Autorin oder den Autor angreifbar machen. Man könnte z. B. die folgende Regel zum Gebrauch der Adposition *wegen* formulieren:

> Das präpositionale wegen ist für alle nominalen Kontexte grammatikalisiert. Das postpositionale wegen ist nur bei determinierten bzw. komplexen Nominalphrasen, nicht aber bei unbekleideten Nomina (Nominalphrasen mit nur einem Nomen) grammatisch.

Vergleichen Sie in (3) und (4), was geschieht, wenn diese Aussage mit introspektiv gewonnenen Daten belegt wird.

(3) *Der neu geborenen Raubkatzen wegen gehen wir in den Zoo.*
(4) **Raubkatzen wegen gehen wir in den Zoo.*

Das sprachliche Beispiel in (4) überzeugt nicht so klar wie die vorigen Beispiele. Mit hoher Wahrscheinlichkeit wird es Betrachterinnen und Betrachter geben, die die als ungrammatisch markierte Äußerung in (4) akzeptabel finden, womit die Argumentation ihr Ziel, die Leserinnen und Leser zu überzeugen, verfehlt hat. Mit Sprachgebrauchsdaten wie Korpusdaten kann man dennoch zeigen, dass Sprecherinnen und Sprecher des Deutschen das postnominale *wegen* ausschließlich bei komplexen Nominalphrasen und allenfalls bei Eigennamen, nicht aber bei einfachen Nomina verwenden. Man benötigt also andere Daten, um gewisse Aussagen glaubhaft belegen zu können. Für eine korpusbasierte Auswertung der Variation von *wegen* siehe Arbeitsaufgabe 1 in Kapitel 3.1.2.18 sowie 3 in Kapitel 4.6.1.

Empirische Daten, die nicht auf Introspektion beruhen, lassen sich weiter nach dem Datengewinnungsprozess untergliedern.

Von elizitierten Daten spricht man, wenn die Daten für das angestrebte Forschungsziel gewonnen wurden. Beispiele hierfür sind aufgenommene Gespräche zum Zweck der Auswertung, geschriebene Aufsätze oder Ergebnisse zu Befragungen.

Nicht-elizitierte und nicht-introspektive Daten sind bereits bestehende Daten, die in der Regel authentischen Produktionskontexten entstammen. Beispiele sind Texte aus Internetforen, gesammelte Zeitungsartikel sowie Romantexte oder Briefe zwischen bestimmten Kommunikationspartnern aus einer bestimmten Zeit.

Experimentelle Daten sind elizitierte Daten, deren Erhebung so kontrolliert wurde, dass gemäß der zu testenden Hypothese oder zu beantwortenden Fragestellung gezielt bestimmte Variablen beeinflusst wurden. Im Sinne des oben formulierten Zusammenhangs zwischen der Komplexität der Nominalphrase und dem postpositionalen Gebrauch von *wegen* könnte man durch eine klar abweichende Reaktion (z. B. in Form gemessener Reaktionszeiten oder durch neurophysiologische Messungen) von Probanden auf Fälle von *wegen* nach einer komplexen Nominalphrase und Fälle von *wegen* nach einem alleine stehenden Nomen zeigen, dass die Varianten Unterschiede in der Verarbeitung hervorrufen, und anhand dieser Zusammenhänge Rückschlüsse auf die Grammatikalität der Varianten ziehen.

Korpusdaten können jeder der drei letztgenannten Kategorien entsprechen. Korpora mit nicht-elizitierten Texten sind diverse Zeitungskorpora wie die syntaktisch annotierten Korpora TIGER (http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html) und TüBa-D/Z (http://www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dz-html) oder sämtliche Korpora, die aus Internetdaten bestehen, sowie sämtliche historische Korpora, für die Textsammlungen aus bestimmten Zeitabschnitten zusammengestellt wurden. Elizitierte Korpora sind all

jene, für die ein Sprachproduktionsanlass geschaffen wurde und bei denen die Produktion mündlicher oder schriftlicher Sprache von Probandinnen und Probanden festgehalten wurde. Beispiele hierfür sind die Korpusprojekte FOLK (http://agd.ids-mannheim.de/folk.shtml) und GeWiss (https://gewiss.uni-leipzig.de/), in denen mündliche Gespräche zwischen bestimmten Personen in bestimmten Situationen aufgezeichnet und weiterverarbeitet wurden, ohne dass jedoch präzise Vorgaben (Stimuli), die gewisse linguistische Parameter kontrollieren, gesetzt wurden. Die Korpora unter den elizitierten Fällen, die als Experimente zu klassifizieren sind, besitzen darüber hinaus mehr oder weniger stark kontrollierte Stimuli für die Sprachproduktion. Ein unterhaltsames Beispiel ist das ›Alcohol Language Corpus – ALC‹ (https://bit.ly/2FjmUx1), in welchem Sprecherinnen und Sprecher ohne und mit bestimmtem Alkoholpegel jeweils dieselben Sprachproduktionsaufgaben bearbeiten.

Klassifizierung des Analyseprozesses: Bislang wurden Unterscheidungen an dem Wesen der Forschungsdaten formuliert, man kann aber auch den Umgang mit den Daten zu ihrer Unterscheidung heranziehen. Steinbach et al. (2007 S. 20 f.) schlagen eine Klassifikation vor, die sich auf die Forschungstätigkeit im Umgang mit Forschungsdaten bezieht. Als nichtintrospektive Möglichkeiten des Forschens werden die Tätigkeiten ›Beobachten‹, ›Befragen‹ und ›Experimentieren‹ vorgeschlagen. Als typisches Beispiel der Beobachtung wird die Erstellung von Gesprächstranskripten mit anschließender Auswertung der Transkripte angeführt. Später werden Korpusdaten der Tätigkeit des Beobachtens zugeordnet. Als die zwei Hauptformen von Befragungen werden Fragebogenstudien und Interviews angeführt. Bezüglich der experimentellen Verfahren werden zwei grundlegende Methoden des Kontrastierens von Messungen beleuchtet, über die eine bestimmte Variable manipuliert werden kann: die Wiederholung von Messungen sowie die Messung über verschiedene Probandengruppen hinweg. Das Beispiel für ersteres Verfahren entspricht dem oben genannten Beispiel des ALC-Korpus (die Messung von Aussprache in Abhängigkeit von Alkoholkonsum bei denselben Sprechern). Als Beispiel für die Manipulation einer Variable über vergleichende Messungen bei verschiedenen Probandengruppen wird die Messung des Lernerfolgs im Grammatikunterricht in Abhängigkeit von zwei verschiedenen Lehrverfahren angeführt. Die Autorinnen und Autoren geben korrekt an, dass die Analyse von Korpora »im Prinzip vom Verfahren her eine Beobachtung« (Steinbach et al. 2007, S. 41) darstellt. Die Korpusdaten selber können aber, wie oben beschrieben, aus Befragungen oder Experimenten hervorgegangen sein.

Qualitative vs. quantitative Verfahren: Häufig wird methodisch zwischen qualitativen und quantitativen Methoden unterschieden. Ist diese Unterscheidung kategorial gemeint, ist sie entschieden abzulehnen. Sie führt nämlich in den meisten Fällen dazu, dass korpusbasierte Forschung fälschlicherweise als quantitative Forschung abgetan wird. Dies liegt daran, dass bei der systematischen Auswertung von Korpora generell mit Häufigkeiten gearbeitet wird. Wer nur auf diesen Aspekt schaut, vernachlässigt allerdings, dass die vorangegangene Aufbereitung der Daten (dies ist die eigentliche Datenanalyse), die Vorbereitung der Auswertung sowie

die Interpretation der Ergebnisse qualitative Analysen darstellen und man mit dem quantitativen Schritt lediglich eine empirische Beweiskraft schaffen möchte. Umgekehrt ist eine rein qualitative Auswertung, solange sie nicht nur der Datenexploration und der Gewinnung erster Arbeitshypothesen dient, schwer denkbar.

Die qualitative Betrachtung eines Phänomens oder einer Variante X kann über die Aussage »X kommt vor« nicht hinausgehen. Doch auch bei sogenannter qualitativer Forschung besteht grundlegend das Bedürfnis, anzugeben, ob eine Beobachtung der Regelfall oder ein Ausnahmefall ist, ob eine gegebene Kategorie zu einem bestimmten Kontext gehört, in dem sie beobachtet wird (also dort häufig auftritt), oder nicht (also im gegebenen Kontext selten oder unsystematisch auftritt). Dies sind quantitative Aspekte, ohne die die Forschung eine Einzelfallbetrachtung bleibt. In diesem Sinne sind Aussagen in sogenannten qualitativer Forschung wie »X tritt in den analysierten Daten häufig auf«, die nicht mit Frequenzwerten untermauert werden, nicht qualitativer, weil ihnen die genaue Angabe der Häufigkeit fehlt; sie sind in ihrem Gehalt einfach vager, unpräziser oder unwissenschaftlicher. Zum Zusammenspiel von qualitativen und quantitativen Methoden in der Korpuslinguistik vgl. auch Lüdeling (2007).

Marginalie: "Rein qualitative Forschung ist schwer vorstellbar

1.4 | Anwendungsbereiche von Korpora

Korpora sind nicht nur in verschiedenen Wissenschaftsbereichen relevant (s. Kap. 1.4.1), sondern besitzen auch allgemeinere Verwendungszwecke (s. Kap. 1.4.2), deren Relevanz mit besserer Zugänglichkeit von Korpora zunimmt.

1.4.1 | Korpora als empirische Grundlage in linguistischen Disziplinen

1.4.1.1 | Korpora in der Grammatikforschung

Korpora bilden den Sprachgebrauch ab – sie lassen Aussagen darüber zu, welche Wörter und Strukturen von Sprecherinnen und Sprechern einer Sprache in bestimmten Situationen verwendet werden, welche vermieden werden bzw. nicht vorkommen und welche Abhängigkeiten zwischen Wörtern, grammatischen Strukturen und außersprachlichen Variablen bestehen. Durch Annotationen in Korpora können grammatische Strukturen quantifiziert und zueinander in Beziehung gesetzt werden, so dass Aussagen über Häufigkeiten, Unterschiede, Wechselwirkungen usw. gemacht werden können. Intuitiv lassen sich Fragen nach Häufigkeitsverteilungen schwer beantworten und die Antworten lassen sich nicht belegen.

Die Grammatikforschung interessiert sich allerdings nicht nur für die sprachlichen Strukturen, die wir nutzen, sondern auch für die Strukturen, die wir nicht nutzen. Hier stoßen wir in der Korpuslinguistik auf ein

grundlegendes Problem: Häufig ist das Ziel der Grammatiker, mögliche Strukturen (z. B. *einige große Biere*) von unmöglichen Strukturen (z. B. *einiges *großen Biere*) abzugrenzen. Das Nicht-Vorkommen einer gegebenen Struktur im Korpus belegt jedoch nicht ihre Unmöglichkeit. Im Korpus nicht vorkommende Strukturen können entweder unmögliche oder mögliche, aber innerhalb der Daten nicht genutzte Ausdrücke darstellen. Dies liegt daran, dass ein Korpus immer eine begrenzte Datenmenge umfasst, von der man nie auf ›die Sprache an sich‹ schließen kann. Deshalb werden Korpora als Beweismittel für die Grammatikalität von Strukturen herangezogen, nicht aber für Ungrammatikalität.

Dies heißt jedoch nicht, dass Nichtvorkommen von Strukturen keine linguistische Evidenz darstellten. Mittlerweile existieren riesige Korpora, so dass das Nichtvorkommen gewisser Strukturen eine relativ hohe Interpretationskraft besitzt.

Beispielhafte Publikationen und Ressourcen aus der grammatischen Forschung: In vielen Arbeiten werden korpusbasiert grammatische Strukturen nachgewiesen, die in der grammatischen Beschreibung des Deutschen bislang als ungrammatisch galten:

- Müller (2005 und 2003), Bildhauer (2011) sowie Müller et al. (2012) weisen nach, dass es völlig kanonische Fälle der doppelten Vorfeldbesetzung im Deutschen gibt und beleuchten die Faktoren, die zur Lizensierung der Struktur doppelt besetzter Vorfelder führt;
- Imo (2008) und Zeng (2016) behandeln Fälle von Modalpartikeln in der Vorfeldposition, die laut der Beschreibung von Modalpartikeln eigentlich ausgeschlossen sein müssten;
- Heylen/Speelman (2003) errechnen anhand syntaktisch annotierter Korpora mit einer multivariaten bzw. multifaktoriellen Analyse die Stärke von fünf verschiedenen Einflussfaktoren auf die Stellung von Subjekt- und Objekt-Nominalphrasen im Mittelfeld des deutschen Satzes. Die Einflussfaktoren sind die grammatische Kategorie, die Länge der Konstituenten in Silben, der Satztyp, in dem die Konstituenten eingebettet sind (Verbzweit- oder Verbletztsatz), die Vorerwähntheit der Konstituenten (neu oder gegeben) und die Belebtheit der Konstituenten. Die Autoren zeigen, dass (wider Erwarten) die grammatische Rolle der Argumente eine eher untergeordnete Rolle spielt und der Satztyp die stärkste Voraussagekraft der fünf gemessenen Faktoren besitzt. Diese Ergebnisse sind interessanterweise mit den Grundannahmen wesentlicher syntaktischer Modelle inkompatibel.

Diese Fälle zeigen, dass die Sammlung authentischer Belege zunächst die Evidenz für bestimmte Konstruktionen und Zusammenhänge liefern kann, die bislang übersehen wurde. Darauf aufbauende Datenanalysen können das grammatische Verständnis schärfen und verändern.

Folgende Ressourcen dienen der korpusbasierten Grammatikforschung; sie werden in Kapitel 3 (zur Korpussuche) sowie Kapitel 4 (zur Auswertung von Korpusdaten) aufgegriffen:

- das Suchsystem COSMAS II des Instituts für Deutsche Sprache in Mannheim für geschriebene Sprache (https://cosmas2.ids-mannheim.de/; s. Kap. 3.2.2)

- das Suchsystem DGD des Instituts für Deutsche Sprache in Mannheim für gesprochene Sprache (https://dgd.ids-mannheim.de/; s. Kap. 3.2.3)
- die syntaktisch annotierten Baumbanken mit deutschen Zeitungstexten: das TIGER-Korpus (https://bit.ly/1cmgyFw) und das TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL) (s. Kap. 3.1.2)

Typische Fragestellungen in der korpusbasierten Grammatikforschung:

1. Passen die in den Standardgrammatiken festgehaltenen Regeln und Tendenzen (noch) zum aktuellen Sprachgebrauch?
2. Welche sprachlichen Strukturen kommen in bestimmten sprachlichen Kontexten häufig vor, welche seltener, welche lassen sich in einer gegebenen Datenmenge nicht nachweisen?
   - Zur Suche von Formen und Strukturen in Korpora s. Kap. 3.
   - Zur Auswertung von Korpussuchen s. Kap. 4.
   - Zur Erstellung normalisierter Frequenzauswertungen s. Kap. 4.3 sowie 4.5.1.
3. Mit welchen Merkmalen geht eine bestimmte Kategorie einher, wovon hängt sie ab?
   - Zur Messung von Korrelationen s. Kap. 4.7.
4. Welche Wörter und Strukturen kommen übermäßig häufig zusammen vor, so dass sie als assoziiert bezeichnet werden können?
   - Zur Messung von assoziierten Elementen s. Kap. 4.6.2.

1.4.1.2 | Korpora in der Varietätenlinguistik

Sprachliche Varietäten sind bestimmte Ausformungen eines gegebenen Standards, z. B. des Standarddeutschen, die sich systematisch beschreiben lassen. Typen von Varietäten sind z. B. Dialekte (Sprache in verschiedenen Sprachräumen), die Sprache bestimmter Gesellschaftsgruppen (sog. Soziolekte), die Sprache in bestimmten kommunikativen Situationen (sog. Register) usw.

Je nachdem, wie wir spezifische Parameter betrachten (Sprachraum, Gesellschaft, Kommunikationssituation etc.), können wir verschiedene Varietäten definieren und beschreiben. Auch die Sprache älterer Sprachstufen und die Sprache von Lernenden einer Mutter-, Fremd- oder Zweitsprache sind ~~zu einem gegebenen Zeitpunkt~~ als Varietäten beschreibbar (s. hierzu Kap. 1.4.1.3 und 1.4.1.4).

Beispielhafte Publikationen und Ressourcen aus der Varietätenlinguistik:

- Auf Douglas Biber geht zum einen die Ausformulierung einer Methodologie für die korpusbasierte Varietätenforschung zurück (vgl. zusammenfassend Biber/Jones 2009, s. auch Kap. 4.4). Er hat sprachliche Variation über verschiedene Varietäten hinweg (Textsorten, Genres und Stile sowie ~~aktuell vor allem~~ Register) anhand von Korpusdaten untersucht und die sprachlichen Variablen, die das Variationsphänomen darstellen, eingehend analysiert (vgl. z. B. Biber 1988 und Biber/Conrad 2009).
- Ein Sammelband, der verschiedene Aspekte von Variation und ver-

schiedene varietätenlinguistische Perspektiven (geographische, sprachvergleichende und registerbezogene) zusammenbringt, ist Szmrecsanyi/Wälchli (2014).
- Eine interessante Korpusressource der dialektalen Mündlichkeit bilden die Dialekt- bzw. Mundartenkorpora, die in der Datenbank gesprochenes Deutsch (DGD, https://dgd.ids-mannheim.de/) verfügbar sind (s. Kap. 3.2.3).

Typische Fragestellungen in der Varietätenlinguistik:
1. Tritt ein Wort oder eine Struktur in einer bestimmten Varietät besonders häufig auf, so dass dieses Merkmal als typisch für die Varietät bezeichnet werden kann?
   - Zum Vergleich von Merkmalen in verschiedenen Korpora s. Kap. 4.5.
2. Zu welcher Variable gehört ein bestimmter sprachlicher Ausdruck, welche Varianten stellt die Sprache noch zur Verfügung und wie sind diese in bestimmten Varietäten repräsentiert?
   - Zur Definition von Variablen und Varianten s. Kap. 4.6.1.

1.4.1.3 | Korpora in der historischen Linguistik

Historische Linguistinnen und Linguisten sind von Natur aus auf die Arbeit mit Korpora angewiesen, denn wenn man ältere Sprachstufen untersuchen möchte, dann bleiben als empirische Basis lediglich Schriftdokumente. In der sprachgeschichtlichen Forschung wurden gesammelte Texte als Grundlage für die linguistische Forschung verwendet, lange bevor die Digitalisierung der Welt eingesetzt hat und massenhaft Textmaterial im Internet zur Verfügung stand. Weil der Korpusbegriff bereits im analogen Zeitalter für die Bezeichnung eben dieser linguistischen Datengrundlage galt, wird er zum Teil heute noch weiter definiert als in anderen Fachbereichen. Wir können aber auch schlussfolgern, dass der Begriff innerhalb der Linguistik seinen Ursprung in der historischen Linguistik hat.

Beispielhafte Publikationen und Ressourcen aus der historischen Linguistik: Heutzutage sind allen historischen Linguistinnen und Linguisten die Vorzüge des digital verfügbaren Korpus bewusst. Folgende wissenschaftliche Beiträge sind dafür bezeichnend.
- Der Sammelband Gippert/Gehrke (2015) spiegelt die aktuellen Trends und Forschungsfragen innerhalb der historischen Linguistik wider. Es werden sowohl die wesentlichen Korpusressourcen für ältere Sprachstände des Deutschen vorgestellt, zu denen u. a. die folgenden Korpusprojekte gehören:
  - Referenzkorpus Altdeutsch (http://www.deutschdiachrondigital.de/, Donhauser 2015),
  - Referenzkorpus Mittelhochdeutsch (http://www.linguistics.rub.de/rem/, Petran et al. 2016),
  - Referenzkorpus Mittelniederdeutsch (http://www.slm.uni-hamburg.de/ren.html, Peters 2017) und
  - Referenzkorpus Frühneuhochdeutsch (http://www.ruhr-uni-bochum.de/wegera/ref/).

Zudem behandelt der Band Forschungsprojekte zu historisch-linguistischen Fragestellungen (vgl. z. B. Coniglios/Schlachters Beitrag zu den Einflussfaktoren der Nachfeldbesetzung im Mittelhochdeutschen, S. 125 f.).

- Die drei genannten Referenzkorpora werden über das Suchsystem ANNIS (http://corpus-tools.org/annis/) verfügbar gemacht (Anfang 2019 waren noch nicht alle Ressourcen verfügbar). Zur Verwendung dieses Systems s. Kap. 3.1.2.
- Das DTA (Deutsches Textarchiv) der Berlin-Brandenburgischen Akademie der Wissenschaften (http://www.deutschestextarchiv.de/) ist wie die drei oben genannten historischen Textkorpora eine Korpusressource für die historische Linguistik. Hier sind Texte aus dem 15. bis 20. Jh. lizenzfrei durchsuchbar. Die Datenmenge ist mit gut 200.000.000 Token (Textwörtern und Satzzeichen) relativ groß (die Angabe stammt von Anfang 2019; der aktuelle Stand kann jederzeit unter der Webadresse http://www.dwds.de/r eingesehen werden). Zur Handhabung des Suchsystems s. Kap. 3.2.1.
- Im Projekt RIDGES (Register in Diachronic German Science, http://hu-berlin.de/ridges) zur Erforschung der Entstehung wissenschaftlicher bzw. fachsprachlicher Register wurde über Jahre hinweg das RIDGES-Korpus (Odebrecht et al. 2017) erstellt, welches über das ANNIS-Suchinterface der Humboldt-Universität zu Berlin frei verfügbar ist (https://hu.berlin/ridges-korpus).

Typische Fragestellungen in der historischen Linguistik:

1. Wie verändern sich Wörter oder syntaktische Strukturen über die Zeit?
2. Welche sprachlichen und außersprachlichen Faktoren spielen beim Sprachwandel eine Rolle?
3. Welche sprachlichen Varianten existieren in einem bestimmten Zeitraum und einem bestimmten Sprachraum?
   - Zur Ermittlung von Schreibvarianten (Anleitungskasten im unteren Kapitelteil) und morphologischen Varianten (Arbeitsaufgabe 1) im Mittelhochdeutschen bzw. Frühneuhochdeutschen s. Kap. 4.6.1.

1.4.1.4 | Korpora in der Spracherwerbsforschung

Auch im Feld der Spracherwerbsforschung werden Korpora traditionellerweise genutzt. Sie werden im Allgemeinen als Lernerkorpora bezeichnet, weil sie Sprachdaten von Lernenden einer Sprache enthalten. Bei der entsprechenden Korpusanalyse möchte man Aussagen über den Erwerb einer Fremd- oder Zweitsprache empirisch fundieren können. Das Ziel ist es deshalb, möglichst große und möglichst gut kontrollierte Sammlungen authentischer Texte von im Spracherwerb befindlichen Lernenden zu sammeln und zu analysieren. Im Fall von Zweit- oder Fremdspracherwerb ist es sinnvoll, Lernerdaten mit vergleichbaren Daten nativer (muttersprachlicher) Sprecherinnen und Sprecher zu vergleichen. Im Fall des muttersprachlichen Erwerbs lassen sich jüngere Sprachlernende mit älteren vergleichen. Daraus sollen jeweils Einblicke

in den Erwerbsprozess ermöglicht werden, die ohne die empirische Methode verborgen bleiben.

Es gibt zwei grundlegende Perspektiven auf die Sprache von Lernenden zu schauen: Entweder analysiert man Abweichungen in der Sprachproduktion der Lernenden hinsichtlich eines Standards bzw. einer Norm (dies wird auch als Fehleranalyse bezeichnet) oder man analysiert Abweichungen zwischen einer Lernergruppe und einer anderen Sprechergruppe mithilfe entsprechender Vergleichskorpora, die sich idealerweise nur in dem Parameter ›Erwerbsstand‹ unterscheiden.

Marginalie: "Fehlerstudien vs. Vergleichsstudien"

Beispielhafte Publikationen und Ressourcen aus der Lernerkorpusforschung:

- Sylviane Granger hat die kurz angerissene Methodologie im Bereich der Auswertung von Lernerkorpora maßgeblich mitgeprägt: Vgl. z. B. zusammenfassend Granger (2008). Hier werden die Verfahren der Fehleranalyse (Englisch: Error Analysis, EA) sowie der kontrastiven Analyse von Lernersprache (Englisch: Contrastive Interlanguage Analysis, CIA) bündig dargelegt.
- Von Granger und ihren Mitarbeiterinnen stammt auch eine wesentliche Korpusressource zum Erwerb des Englischen als Fremdsprache, das ICLE-Korpus (International Corpus of Learner English, https://bit.ly/2HzNfdE, Granger et al. 2009). Von der Forschergruppe wurde u. a. eine umfassende Liste weltweiter Lernerkorpora veröffentlicht: https://bit.ly/2Cw2WOR. Publikationen des Teams (meistens zu den ICLE-Korpusdaten) finden sich unter der Webadresse https://bit.ly/2Fkn3QK).
- Für den Bereich Deutsch als Fremdsprache ist das Lernerkorpus Falko (https://hu.berlin/falko) eine frei verfügbare Ressource, zu der die komplexe Erstellungsmethodik in Form einer umfassenden Korpusdokumentation im Internet verfügbar ist (https://hu.berlin/falko-handbuch, Reznicek et al. 2012). Wesentliches zum Aufbau des Korpus und den Forschungsmöglichkeiten, die sich daraus ergeben, sind in Lüdeling et al. (2008) veröffentlicht. Das Korpus ist über eine Instanz des ANNIS-Suchinterfaces an der Humboldt-Universität zu Berlin frei zugänglich (https://hu.berlin/annis-falko).
- Mehrere Korpora, die für allgemeinere oder andere spezifische Zwecke erstellt wurden, enthalten zu gewissen Anteilen sprachliche Beiträge von im Spracherwerb befindlichen Sprecherinnen und Sprechern. Beispiele hierfür sind das GeWiss-Korpus zur gesprochenen Wissenschaftssprache (https://gewiss.uni-leipzig.de/, Frandrych et al. 2017) oder das BeMaTaC-Korpus zu gesprochenen Maptask-Dialogen (https://hu.berlin/bematac, s. auch Kap. 4.5.3, Sauer/Lüdeling 2016).

Typische Fragestellungen in der korpusbasierten Spracherwerbsforschung:

1. Welche Kategorien der zu erwerbenden Sprache stellen im Erwerbsprozess eine Hürde dar? (Bekannte Erwerbsphänomene des Deutschen sind z. B. die variierende Verbstellung und die Nominalflexion.)
2. Wie äußern sich die Erwerbsprobleme im Sprachgebrauch der Lernenden?

3. Welche didaktischen Implikationen ergeben sich aus diesen Erkenntnissen (wie kann man Lernenden den Zugang zur Zielgrammatik erleichtern)?
   - Hierzu müssen in der Regel Verwendungshäufigkeiten bestimmter grammatischer Kategorien zwischen Lernenden und Nicht-Lernenden verglichen werden. Für entsprechende Beispiele der Auswertung von Lernerkorpora s. Kap. 3.1.2.27 und 4.5.4.

1.4.1.5 | Korpora in der Lexikographie

Die Lexikographie ist die Wissenschaft von der Erstellung von Wörterbüchern zu Einzelsprachen oder bestimmten Varietäten. Hierbei geht es nicht nur um die Frage, welche Wörter in ein bestimmtes Lexikon aufgenommen werden sollen und welche nicht, sondern fast jedes Lexikon bietet über das eigentliche Wortverzeichnis hinaus Beschreibungen bzw. Erklärungen der Einträge. Je nach Zweck des Wörterbuchs kann dies Worterklärungen bzw. -definitionen umfassen. Lexikographen sind heute im Allgemeinen auf große Korpora angewiesen, da diese die verlässlichste Quelle über den Sprachgebrauch und quantitative Verteilungen bieten.

Welche Korpusressourcen Lexikographen konkret verwenden, hängt vom Ziel der Zusammenstellung ab, und prinzipiell kann jedes Korpus für lexikographische Zwecke herangezogen werden. Vor allem eignen sich jedoch große, flexibel kompilierbare (zusammenstellbare) Korpora wie die in COSMAS II des Instituts für Deutsche Sprache verfügbaren Korpusressourcen (http://www.ids-mannheim.de/cosmas2/; s. Kap. 3.2.2 für die Verwendung dieses Suchsystems) für lexikographische Zwecke.

Beispielhafte Korpusressourcen in lexikographischem Kontext:

- Ein wichtiges lexikographisches Projekt, das in seinem Ergebnis sowohl ein online verfügbares Wörterbuch als auch eine Quelle für die Erfassung von Lexika darstellt, ist das DWDS (Digitales Wörterbuch der deutschen Sprache, http://www.dwds.de/). Zur ausführlichen Vorstellung der Such- und Auswertungsmöglichkeiten in diesem System s. Kap. 3.2.1.
- Zweisprachige Wörterbücher zu Übersetzungszwecken stellen spezifische Anforderungen an die Herausgeberinnen und Herausgeber, weil sich Wörter (Wortformen) nicht ohne spezifische Bedeutungskontexte übersetzen lassen. Das online verfügbare mehrsprachige Übersetzungsportal Linguee (http://www.linguee.de) ist ein Beispiel für ein korpusgestütztes System, welches verschiedene Übersetzungs- bzw. Parallelkorpora nutzt, vor allem aus dem Kontext der Europäischen Union, wo z. B. Parlamentssitzungen und Gesetzestexte in sämtliche Sprachen übersetzt werden. Solche Daten können, soweit verfügbar, für jeden Suchbegriff als Belegdaten eingesehen werden.

Typische Fragestellungen in der Lexikographie sind:

1. Welche Wörter werden in bestimmten sprachlichen Kontexten häufig verwendet, welche sind selten und können deshalb als markiert eingestuft werden?
2. Wie viele und welche Lesarten hat eine bestimmte Wortform?
3. Welche ggf. besonderen bzw. unregelmäßigen Formen gehören bei einem flektierbaren Wort zum Flexionsparadigma?
   - Diese drei Fragen implizieren zunächst systematische Suchen nach Wort- und Grundformen. Siehe hierzu Unterkapitel von Kap. 3 wie z. B. Kap. 3.1.2.2, 3.1.2.5 oder 3.2.
4. Welche Wörter treten häufig gemeinsam auf (und bilden somit sog. Kollokationen)?
   - Für die Ermittlung miteinander assoziierter Wörter s. Kap. 4.6.2.

1.4.2 | Weitere Anwendungszwecke von Korpora

Außerhalb der verschiedenen linguistischen Forschungsdisziplinen können Korpora auch in anderen, häufig praktischeren Bereichen ihre Anwendung finden. Die Nutzung von Korpora scheint vor allem durch technische Hürden beschränkt zu sein: Während routinierte Nutzerinnen und Nutzer von Korpora diese in allen möglichen Kontexten des Umgangs mit Sprache verwenden (s. u.), berichten noch nicht in die Disziplin Eingeweihte von erheblichen Start- bzw. Zugangsschwierigkeiten. Es ist stark anzunehmen, dass durch die Absenkung von Zugangshürden bei der Verwendung korpuslinguistischer Ressourcen Nicht-Korpuslinguistinnen und -linguisten Korpora deutlich stärker nutzen würden. Im Folgenden sind Anwendungsfelder aufgelistet, die nicht nur Forschende interessieren dürften, sondern z. B. auch von Lehrerinnen und Lehrern, Deutschlernenden und professionellen oder nicht professionellen Schreibenden genutzt werden können.

Korpora als Hilfe in der Texterstellung: Häufig fragt man sich bei der Erstellung eigener Texte, ob gewisse Ausdrücke geläufig sind oder welche Wörter gängige Kollokationen zu bestimmten anderen Wörtern sind. Hier sind Korpora eine nützliche Ressource: Sofern man sich mit der Bedienung auskennt, sind Korpora schnell verfügbar und können objektiver und zuverlässiger Auskunft über den Sprachgebrauch geben als Menschen.

Korpora als Belegressource: Der offensichtlichste und zugleich einfachste Anwendungszweck für Korpora ist, dass sie eine Quelle für authentische Sprachbelege sein können. Da die Grammatikalität graduell ist und keineswegs eine mittels einer scharfen Trennlinie auszumachende Dichotomie von grammatischen und ungrammatischen Strukturen darstellt, ist es für die Glaubwürdigkeit linguistischer Beiträge von erheblichem Nutzen, exemplarische Belege für grammatische Konstruktionen, Verwendungskontexte usw. nicht selber zu konstruieren, sondern von Dritten einzuholen. Die relevanten sprachlichen Strukturen sollten nicht für den Zweck der linguistischen Betrachtung hervorgebracht worden sein, sondern authentischen sprachlichen Kontexten entstammen. Essen-

ziell hierfür ist das Wissen darüber, wie man bestimmte Konstruktionen gezielt finden kann, vor allem wenn es sich um abstrakte syntaktische Konstruktionen handelt, die nicht formbasiert abgebildet werden können. Siehe Kapitel 3 für die Suche in verschiedenen Korpussuchsystemen mit unterschiedlichen Anfragesprachen nach vielzähligen linguistischen Kategorien und Mustern.

Werden viele Belege zu einem bestimmten Phänomen korpusbasiert zusammengestellt, so entsteht eine Belegsammlung bzw. Datensammlung, die gemäß der in Kapitel 1.2 vorgestellten Definition(en) nicht mehr als Korpus bezeichnet werden können. Existierende Sammlungen dieser Art sind z. B. das in Müller (2013) zusammengestellte schriftsprachliche Datenmaterial zur (scheinbar) doppelten Vorfeldbesetzung im Deutschen (diskutiert in Müller 2005 und 2003 sowie Müller et al. 2012), ganz ähnlich die Sammlung von Stellungsvariationen im Vorfeldbereich des gesprochenen Deutsch in Schalowski (2015), S. 70 f., oder die in der Übersetzungsressource ›Linguee‹ (http://www.linguee.de/) entstehende Übersicht bei der Auswertung von Übersetzungsbegriffen.

Korpora in der Lehre: Im Unterricht haben Korpora vielseitige Verwendungsmöglichkeiten. Zuallererst können sie auch in diesem Kontext dazu verwendet werden, authentische Textbelege zu liefern, die in Unterrichtsmaterialien als Illustrationsbeispiele dienen oder in Prüfungsaufgaben die Grundlage für linguistische Analysen sind.

Vor allem in der Fremdspracherwerbsforschung existieren linguistische und didaktische Forschungsbereiche, die sich mit computergestützten Lehr- und Lernmethoden befassen, wobei Korpora häufig eine wichtige Rolle spielen. Zwei wichtige Fachbezeichnungen, unter denen theoretische und praktische Methoden diskutiert werden, wie Korpora als Lehr- und Lernressource genutzt werden können, sind CALL (*computer-assisted language learning*) sowie DDL (*data-driven learning*). Hierbei wird, wie eingangs erwähnt, der Zugang zur Nutzung von Korpora immer wieder als ein zentraler Aspekt erwähnt und gleichzeitig in vielen Studien gezeigt, dass korpusunterstütztes Lehren und Lernen im Klassenraum und im Selbststudium einen positiven Effekt auf den Erwerbsverlauf hat (vgl. Breyer 2009 für einen Ansatz, der Lehrerinnen und Lehrer als die zentrale Instanz für die Vermittlung von Korpora an eine breite Nutzergemeinschaft ansieht und den Effekt der Korpusnutzung im Sprachunterricht empirisch untersucht; vgl. Imo/Weidner 2018 für den Einsatz von Korpora im DaF- und DaZ-Unterricht).

Korpora als Trainings- bzw. Lernressource für automatische Anwendungen: Ein riesiger Forschungsbereich ist das Erstellen und Verbessern von Anwendungen zur automatischen Verarbeitung von Sprache. Der Fachbereich wird auch ›NLP‹ (Natural Language Processing) genannt. Um Programme die Analyse linguistischer Kategorien wie Wortarten, Textsorten oder semantische Klassen lernen zu lassen, bedarf es kontrollierter Datensammlungen, die bereits Analysen enthalten, also in aller Regel Korpora. Eine englischsprachige Zusammenstellung der klassischen computerbasierten Methoden im NLP-Bereich, die in diesem Buch nicht detailliert thematisiert werden können, bieten Clark/Fox/Lappin (2010).

1.5 | Welches ist das passende Korpus für meine persönliche Fragestellung? – Korpora und ihre ›Repräsentativität‹

Eine der am häufigsten gestellten Fragen in Einführungsveranstaltungen zur korpuslinguistischen Methodik ist die nach der Beschaffenheit des Korpus, das für eine gegebene Forschungsfrage herangezogen werden soll. Meistens wird in diesem Kontext nach der Größe und der Zusammensetzung der Korpusdaten gefragt. Man kann die vielen verschiedenen Fragen auf zwei wesentliche reduzieren:

1. Wann ist ein Korpus groß genug, um repräsentativ zu sein?
2. Wie viele verschiedene Varietäten muss ein Korpus enthalten, um repräsentativ zu sein?

Wer hierauf eine positive, konkrete und abschließende Antwort erwartet, wird leider enttäuscht. Dass es keine Pauschalantwort auf diese Fragen geben kann, zeigt folgende Vorstellung: Was würde man antworten, wenn jemand fragt: »Was muss ich lesen, damit ich im Fach XY Bescheid weiß?« Wahrscheinlich würde man etwas antworten wie: »Das hängt vom aktuellen Wissensstand ab und von dem konkreten Ziel des Wissensgewinns.« Im korpuslinguistischen Kontext ist dies nicht anders: Es ist zum einen von entscheidender Bedeutung, welche Zwecke die Korpusnutzung erfüllen soll. Korpora können auf verschiedene Weisen informativ sein, z. B. können sie zur Generierung von Hypothesen oder zu ihrer Überprüfung verwendet werden. Anders ausgedrückt, können quantitative Verfahren in der Linguistik grundlegend dem Beschreiben, dem Erklären oder dem Vorhersagen linguistischer Sachverhalte dienen (vgl. Gries 2008, S. 10). Jeweils gelten dabei andere Ansprüche an die Daten und Auswertungsmethoden.

Die Menge und Beschaffenheit von Korpusdaten ist also hochgradig abhängig von der Forschungsfrage, die man an sie stellt. Jede einzelne Forschungsfrage bedingt eine eigene optimale Datengrundlage. Und ohne eine konkrete Forschungsfrage ist die Verwendung von Korpora zwecklos. Die Suche nach dem perfekten Korpus ist also eine Einzelfallbetrachtung. Gemäß diesem Einzelfall muss geklärt werden, welche sprachliche Datenmenge – statistisch sagt man ›welche Grundgesamtheit an Sprachdaten‹ – idealerweise untersucht werden müsste. Hat man hierfür eine Antwort, kann man Überlegungen darüber anstellen, wie eine untersuchbare Teilmenge dieser Gesamtmenge – statistisch sagt man ›eine Stichprobe aus der Grundgesamtheit‹ – aussehen müsste, so dass sie möglichst repräsentativ für die Grundgesamtheit ist, über die etwas herausgefunden werden soll.

Fallbeispiel: Stellen Sie sich vor, Sie wollten ›die Sprache Goethes‹ repräsentativ untersuchen und ein Wörterbuch mit Häufigkeiten der Wörter, die Goethe verwendet hat, erstellen. Wenn Sie wirklich ›die Sprache Goethes‹ meinen, können Sie die Grundgesamtheit nicht angeben, denn dies würde sämtliche Äußerungen des Dichters mit einschließen (u. a. mündliche), die nicht veröffentlicht wurden. Man kann ›die Sprache Goe-

thes‹ definieren als ›dasjenige von Goethe, was jemals veröffentlicht wurde‹. Wenn man das Gesamtwerk, sämtliche Briefe des Dichters usw. als zu umfangreich erachtet, um es im Gesamten auszuwerten, muss man ermitteln, zu welchen Anteilen die einzelnen Genres und Beiträge im Gesamtwerk vertreten sind, um dann eine Abwägung darüber anzustellen, wie genau eine Stichprobe daraus diesen Proportionen gerecht werden kann.

Ein alternativer Begriff bzw. ein zur Repräsentativität alternatives Konzept ist das der Referenz: Ein Referenzkorpus (z. B. für das Deutsche, für das gesprochene Deutsch, für das Althochdeutsche usw.) muss nicht repräsentativ sein, sondern die folgenden Ansprüche erfüllen: Es sollte kontrolliert zusammengestellt sein, so dass wesentliche Parameter (wie Textsorte, Alter der Texte, Themen usw.) über die Gesamtdaten gesehen ausgewogen bzw. homogenisiert sind oder anhand von Metadaten berücksichtigt werden können und je nach Auswertungsziel homogenisiert werden können. Deutschsprachige Korpora mit diesem Anspruch sind

- die historischen Referenzkorpora aus der Verbundsinitiative DDD (Deutsch diachron digital, http://www.deutschdiachrondigital.de/projekt/) Referenzkorpus Altdeutsch (http://www.deutschdiachrondigital.de/), Mittelhochdeutsch (http://www.linguistics.rub.de/rem/), Mittelniederdeutsch (http://www.slm.uni-hamburg.de/ren.html) und Frühneuhochdeutsch (http://www.ruhr-uni-bochum.de/wegera/ref/), die nach Zeiträumen und Dialektgebieten homogenisiert sind, wenn es die Datenlage erlaubt,
- das Korpus ›DeReKo‹ (deutsches Referenzkorpus, http://www1.ids-mannheim.de/kl/projekte/korpora) des Instituts für Deutsche Sprache, dessen verschiedenen Textarten im Suchsystem COSMAS2 (http://www.ids-mannheim.de/cosmas2/) kontrolliert werden können,
- das Korpus ›FOLK‹ (Forschungs- und Lehrkorpus gesprochenes Deutsch, http://agd.ids-mannheim.de/folk.shtml, Deppermann/Schmidt 2014), dessen gesprochensprachliche Dialoge anhand verschiedener registerbezogener Parameter kontrolliert werden können.

1.6 | Ein Entscheidungsbaum zur Verwendung von Korpora

Aus den vorangegangenen Überlegungen ergibt die in Abb. 1.1 dargestellte Grundanleitung zur Verwendung von Korpora. Auch wenn Korpora als modernes Instrument der Evidenzgewinnung ein gewisses Prestige genießen, sollten sie nicht einfach um dieses Prestiges willen, sondern überlegt erstellt und genutzt werden. Nur wenn ein Korpus zu einer linguistischen Fragestellung Evidenz beitragen kann und der Aufwand, den seine Erstellung und Nutzung kostet, gerechtfertigt ist, sollte eine korpuslinguistische Herangehensweise gewählt werden. Lesen Sie den Entscheidungsbaum (s. Abb. 1.1) von oben nach unten und behalten Sie die Fragen bei der Bearbeitung von Kapitel 2 sowie 3 im Hinterkopf.

Kapitel 2 führt in den Aufbau von Korpusressourcen ein. Hierdurch

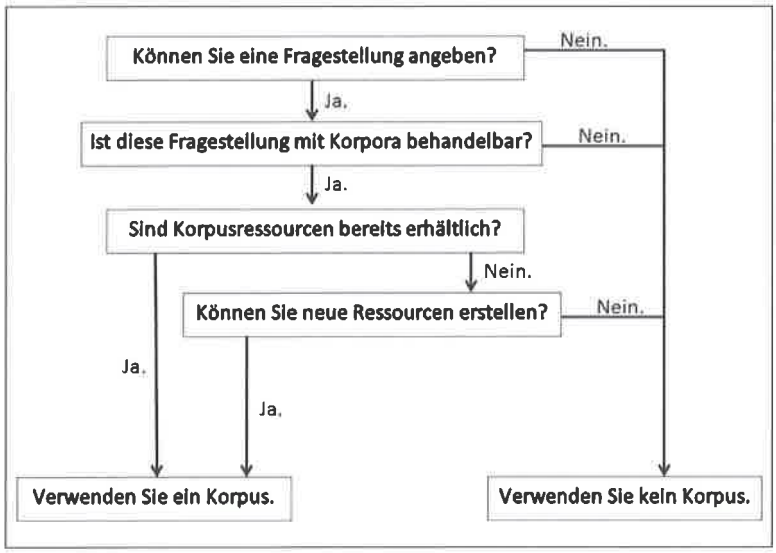


Abb. 1.1: Entscheidungsbaum für die Verwendung von Korpora

sollten die Leserinnen und Leser ein Gefühl dafür bekommen, wie aufwändig die Erstellung unterschiedlicher Korpusressourcen mit verschiedenen automatisch, halbautomatisch oder manuell erstellten Annotationen ist. Kapitel 3 behandelt die Suche in eigens erstellten sowie bereits vorhandenen Korpusressourcen. Kapitel 4 stellt die wesentlichen methodischen Schritte bei der Korpusauswertung vor, wobei stets Anwendungsfälle genannt werden. Anhand dieses Kapitels sollte nachvollziehbar werden, welche Fragestellungen an Korpora gestellt werden können und welche nicht.

2 Praxisteil I: Erstellung und Aufbau von Korpora

2.1 Datenakquise: Gewinnung von Korpus-Primärdaten
2.2 Vom geschriebenen Text zum Korpus: Erstellung von Textkorpora
2.3 WebLicht: eine Online-Plattform zur automatischen Verarbeitung von Korpusdaten
2.4 Vom gesprochenen Text zum Korpus: Erstellung von Gesprächskorpora
2.5 Evaluation von Korpusannotationen
2.6 Daten über die Daten: Annotation von Metadaten

Bevor in Kapitel 3 und 4 die Auswertung von Korpusdaten anhand bestehender Ressourcen behandelt wird, wird in diesem Kapitel besprochen, wie man Korpusdaten selber erstellt. Die Korpuserstellung lässt sich in die folgenden drei Bereiche unterteilen:

1. die Datenakquise (die Gewinnung von Primärdaten, s. Kap. 2.1),
2. die Annotation (die Anreicherung der akquirierten Daten mit linguistischen Informationen; hierzu gehört auch die Vergabe sog. Metadaten; s. Kap. 2.2–2.6),
3. die Bereitstellung der Korpusdaten (für etliche Beispiele öffentlich bereitgestellter Korpusdaten s. Kap. 3).

Marginalie: "Teilbereiche für die Korpuserstellung"

2.1 | Datenakquise: Gewinnung von Korpus-Primärdaten

Der Grundstein für die Erstellung eines jeden Korpus sind die eigentlichen, unverarbeiteten Textdaten (Primärdaten).

Definition

> Als **Primärdaten** werden diejenigen Daten interpretiert, die der gesamten Korpusaufbereitung zugrunde liegen. Somit sollte gelten: Die Primärdaten sind der Untersuchungsgegenstand, den ein Korpus mit sich bringt.

Häufig liegt auf der Hand, welche Daten des Korpus als Primärdaten zu gelten haben: Im Fall eines Korpus handschriftlich verfasster Liebesbriefe müssen die eigentlichen Briefe als die Primärdaten aufgefasst werden. In den meisten Fällen liefern die existierenden Korpora die Primärdaten gar nicht mit, sondern enthalten als ›primärste‹ Korpusebene eine tokenisierte und somit eine weiterverarbeitete Fassung der dem Korpus zugrunde liegenden Daten (s. Kap. 2.2.5). Ein Liebesbriefkorpus wird im Regelfall auf

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_2

einer abgetippten und tokenisierten Fassung der ursprünglichen Briefe aufbauen. Die Primärdaten zu zeigen, würde in diesem Fall bedeuten, Scans der Originaldokumente mitzuliefern. In vielen Fällen können die Primärdaten aber gar nicht mehr erfasst werden (Liebesbriefe können z. B. vernichtet und nur noch als Abschriften erhalten sein). Häufig ist es auch Auslegungssache, welche Texte als die Primärdaten eines Korpus zu gelten haben. So ist z. B. strittig, ob im Fall eines Korpus aus Reden, die schriftlich vorbereitet wurden, die tatsächlich gehaltenen (mündlichen) Reden, die vorbereiteten Schrifttexte oder beide Quellen die Primärdaten sind. Da je nach Forschungsziel die Antwort unterschiedlich ausfallen muss, ist das Konzept der Primärdaten in der Praxis grundsätzlich problematisch. Dennoch ist der Begriff etabliert und dient der Verständigung über die Beschaffenheit bestimmter Korpora.

Die Primärdaten können medial schriftlich oder medial mündlich vorliegen – je nach dem Untersuchungsgegenstand des Korpus. Die Aufbereitung schriftlicher Sprachdaten ist dabei in aller Regel einfacher, weshalb hier zunächst der Aufbereitungsweg für schriftliche Korpusdaten beschrieben wird und dieser dann um zusätzliche Schritte der Verarbeitung mündlicher Daten erweitert wird. Hierbei wird sich zeigen, dass bei der Aufbereitung schriftlicher und mündlicher Sprachdaten viele wesentliche Verarbeitungsschritte dieselben sind. Vereinfacht ausgedrückt, kommen bei mündlichen Sprachdaten lediglich gewisse Aufbereitungsschritte hinzu.

Die Aufbereitung der Primärdaten wird sowohl bei schriftlichen als auch bei mündlichen Sprachdaten durch zwei Szenarien bestimmt:

1. Die Primärdaten existieren bereits irgendwo (im Internet, auf gedrucktem Papier, in Form von Aufzeichnungen mündlich oder schriftlich abgenommener Prüfungen o. Ä.).
2. Die Primärdaten müssen erst erhoben (elizitiert) werden.

Im Falle von 1. kann dennoch ein gewisser Aufwand vonnöten sein, die zu analysierenden Texte zusammenzustellen und in ein homogenes Datenformat zu bringen. Doch in aller Regel birgt die Aufbereitung bereits vorhandener Textdaten deutlich weniger Arbeitsaufwand als die Erhebung von Texten selbst mit anschließender Datenaufbereitung. Für einige Fachbereiche und Forschungsbelange liegt per se der einfachere Fall vor: Historische Textdaten können nicht erhoben werden, sondern sind entweder existent oder nicht. Im Fall von computerbasierter Kommunikation (die Sprache in bestimmten Internetbereichen, z. B. E-Mails, Forentexten, auch in bestimmten Messengerdiensten usw.) liegt es auf der Hand, dass man auf existente Ressourcen zurückgreift. Dasselbe gilt für Zeitungssprache: Sämtliche moderne Zeitungskorpora bauen auf bereits digital vorliegenden Daten von Zeitungsredaktionen auf, und auch bei der Erstellung von Korpora traditioneller Textgenres (Romane, Lyrik, Fachsprachengenres usw.) kann häufig auf digital vorhandenes Textmaterial zugegriffen werden.

Im Hinblick auf 2. sollte die Regel gelten, dass wegen des zusätzlichen Aufwandes nur dann neue Primärdaten erhoben werden, sofern adäquates Textmaterial noch nicht vorhanden ist. Dies ist bei der Aufbereitung

von Daten medialer Mündlichkeit häufiger der Fall, weil medial mündliche Kommunikation flüchtiger ist und selten konserviert wird. Deshalb haben Projekte zur Untersuchung mündlicher Register wie dem Kiezdeutschen (http://www.kiezdeutschkorpus.de/de/), der gesprochenen Wissenschaftssprache (https://gewiss.uni-leipzig.de/) sowie etlicher Mundartenkorpora und Registerkorpora, die in der »Datenbank für gesprochenes Deutsch« (DGD) gespeichert sind (https://bit.ly/2HxXBuO), Aufnahmen von Sprecherinnen und Sprechern elizitiert und anschließend aufbereitet. Dasselbe gilt für diverse Korpora zum Erstspracherwerb, z. B. dem internationalen Korpus für Kindersprache CHILDES (https://childes.talkbank.org/), und für sämtliche Fremd- und Zweitspracherwerbskorpora wie das Falko-Korpus (http://hu-berlin.de/falko/). Die einzige systematische Ausnahme bilden Korpora, anhand derer man Phänomene der Mündlichkeit untersuchen kann, deren Primärdaten allerdings auf bereits verschriftlichter Rede beruhen, wie im Fall verschiedener Parlamentsredenkorpora (vgl. bspw. die im »Deutschen Referenzkorpus« (DeReKo, http://www1.ids-mannheim.de/kl/projekte/korpora) bzw. in COSMAS II verfügbaren »Plenarprotokolle« des Instituts für Deutsche Sprache in Mannheim: https://bit.ly/2UQUz7O). Sämtliche Korpora, deren Primärdaten elizitiert wurden, müssen als linguistische Experimente verstanden werden. Für sie muss wie für alle Experimente die Frage beantwortet werden, wie man am geeignetsten diejenigen sprachlichen Merkmale elizitiert, die untersucht werden sollen. In diesem Lehrbuch kann diese Problematik nicht eingehend behandelt werden; gute Einführungen in die experimentelle Linguistik finden sich aber häufig in psycholinguistischen Lehrbüchern (vgl. z. B. Dietrich/Gerwien 2017, S. 12 f.).

Aus beiden groben Szenarien – dem Zusammenstellen vorhandener Sprachdaten und dem Elizitieren selbiger – sollen hinsichtlich der korpuslinguistischen Weiterverarbeitung Dateien hervorgehen, die mithilfe von Computerprogrammen eingelesen und weiterverarbeitet werden können. Für schriftliche Daten sind im .txt-Format gespeicherte Texte bzw. solche Dateiformate ideal, die problemlos in ein reines Textformat überführt werden können. Für mündliche Sprachdaten eignen sich möglichst unkomprimierte bzw. hochauflösende Audiodateiformate wie das .wav-Format bzw. solche Formate, die sich problemlos in dieses Format konvertieren lassen. In den anschließenden Kapiteln wird zunächst der Aufbau von Korpora behandelt, deren Grundlage geschriebene Texte sind. Ab Kapitel 2.4 wird die Erstellung von Korpora besprochen, die auf mündlich kommunizierter Sprache basieren (sog. Gesprächskorpora).

2.2 | Vom geschriebenen Text zum Korpus: Erstellung von Textkorpora

Je nach Forschungsfrage können zusätzlich zum sprachlichen Inhalt textstrukturelle (Absätze etc.), typographische (Schriftart und -größe, Schriftformatierung wie Fettdruck usw.) und allgemein grafische Gegebenheiten (Abbildungen etc.) eine Rolle spielen. Hier treffen wir auf die erste we-

sentliche Verarbeitungshürde: Fast alle manuellen und automatischen Annotationsprogramme erlauben nur eine bestimmte Zeichenkodierung und erkennen keine spezifischen Zeichenformatierungen, Textkodierungen usw. Das bedeutet, dass sämtliche Schriftvariationen und individuellen Textstrukturmuster herausnormalisiert werden, wenn wir die Textdaten einlesen und weiterverarbeiten (s. Kap. 2.2.4 zum Normalisierungsprozess). Wenn man jedoch vor der Weiterverarbeitung weiß, welche Merkmale in den Ursprungsdaten für spätere Analysen wichtig sind und bewahrt werden sollen, kann man entsprechend darauf eingehen: Nehmen wir an, wir wollen zwei Texte hinsichtlich diverser linguistischer Merkmale vergleichen, inklusive der Verwendung von Absätzen, Unterstreichungen und Fettmarkierungen. Dann können diese drei Merkmale als Annotationen in der Datenaufbereitung berücksichtigt werden. Vergleichen Sie Kapitel 2.2.3 für die Annotation solcher schrift- und textbezogener Merkmale.

2.2.1 | Annotation (Begriffsklärung)

Definition

> Die **Annotation** von Daten bedeutet, die dem Korpus zugrunde liegenden Daten (Primärdaten bzw. bereits weiterverarbeitete Primärdaten) mit weiteren Informationen anzureichern.

Die Anreicherung durch Annotationen erfolgt im korpuslinguistischen Kontext immer durch eine streitbare und fehleranfällige Interpretation der zu annotierenden Daten, denn keine Zuweisung von Informationen erfolgt rein mechanisch und ohne Interpretationsspielraum. Sämtliche der folgenden Annotationsarten bringen dies zum Ausdruck.

Der Begriff Annotation bezieht sich weder auf einen bestimmten Ausgangstyp von Daten (es kann im Prinzip jedes sprachliche Signal annotiert werden und auch Annotationen können weiter annotiert werden) noch auf eine bestimmte Art der Interpretation. Dies wird vor allem in den folgenden Kapiteln zur Normalisierung (s. Kap. 2.2.4 sowie 2.4.10) und zur Transkription gesprochener Sprache (s. Kap. 2.4.1) relevant, die hier als spezifische Annotationstypen definiert werden.

Zu einer Annotation gehören immer bestimmte Richtlinien ~~[ein Regelwerk]~~, die den Annotationsprozess konsistent machen sollen, also bei gleichartigen Fällen zu denselben Entscheidungen führen sollen. Wenn Sie z. B. Sätze annotieren wollen, müssen Sie Regeln formulieren, wann etwas als ein Satz anzusehen ist. Man kann auch sagen: Sie müssen ›harte‹ Definitionen formulieren, die möglichst wenig Zweifelsfälle erlauben.

Annotieren bedeutet auch immer das Zuweisen von Kategorien bzw. Merkmalsklassen. Bei der Annotation von Sätzen können Sie genau eine Kategorie, nämlich ›Satz‹, zuweisen oder mehrere Kategorien für verschiedene Typen von Sätzen (Hauptsatz, Nebensatz, Verberstsatz, Komplementsatz, Deklarativsatz usw.) vergeben. Die Bezeichnung dieser Ka-

tegorien nennen wir aus dem Englischen kommend Tags (Singular: das Tag). Häufig, aber nicht notwendigerweise, werden Tags in Form von Kürzeln angegeben, um den Prozess des Aufschreibens und Abfragens der Kategorien zu erleichtern. Sie können z. B. »S« für »Satz« vergeben. Ist die Anzahl von Tags endlich, so sprechen wir von einem Tagset. In dem Satz-Annotationsszenario könnten Sie z. B. alle Fälle von Sätzen im Deutschen in »Hauptsatz« (HS), »Nebensatz« (NS) und »Pseudosatz« (PS, z. B. bei Überschriften und bestimmten Einwortäußerungen) untergliedern und hätten somit ein Tagset von drei Kategorien. Bei unendlich vielen Möglichkeiten bei der Kategorisierung können wir kein festes Tagset angeben. Dies geschieht z. B. bei der Lemmatisierung (s. Kap. 2.2.6.1), bei welcher man nur Regeln zur Bildung von Grundformen aus flektierten Formen formuliert, aber unendlich viele Grundformen möglich sind (bei einer absolut unproduktiven Sprache wäre das Tagset für den Annotationsprozess der Lemmatisierung das Lexikon aller Grundformen dieser Sprache).

2.2.2 | Annotation und Annotationswerkzeug – der EXMARaLDA-Partitureditor als Beispiel eines variabel einsetzbaren Annotationsprogramms

Annotationen sind meistens gebunden an spezifische Programme oder Nutzeroberflächen, sogenannte Annotationswerkzeuge. Hierbei handelt es sich um Software, die (annotierte oder nicht annotierte) Korpusdaten in einem spezifischen Format einliest, mittels menschlich gesteuerter oder automatischer Annotationsschritte diese Daten mit Annotationen anreichert und zuletzt die annotierten Daten in einem spezifischen Format wieder ausgibt. Einige Annotationswerkzeuge sind in der Lage, verschiedene Datenformate einzulesen und auszugeben. In den noch folgenden Kapiteln zu verschiedenen Annotationen werden bestimmte Programme vorgestellt, die die jeweils besprochenen Annotationen ermöglichen. EXMARaLDA (http://www.exmaralda.org, Schmidt/Wörner 2014) ist ein Annotationsprogramm, welches inhaltlich nicht auf bestimmte Arten von Annotationen festgelegt ist. Eine spezifische Aufgabe, zu der EXMARaLDA sehr häufig verwendet wird, ist das Transkribieren von Audio- oder Videodaten, die mit den Transkriptionen im Programm verwaltet und zusammengebracht werden (s. Kap. 2.4.2 für die Verwendung von EXMARaLDA als Transkriptionswerkzeug).

Das flexible Annotationsprogramm EXMARaLDA ist ein häufig verwendetes Werkzeug hauptsächlich zur manuellen Transkription und Annotation kleinerer Mengen von Korpusdaten. In Anlehnung an die schriftliche Aufzeichnung mehrstimmiger Musik durch sogenannte Partituren wird es als ›Partitureditor‹ bezeichnet – während sich einer musikalischen Partitur beliebig viele Stimmen hinzufügen lassen, können dem Editor beliebig viele Beschreibungsebenen, die sich auf dieselben Primärdaten beziehen, hinzugefügt werden. Der ›Erfolg‹ dieses Annotationswerkzeugs liegt erstens darin, dass es frei verfügbar ist und inhaltlich auf keine bestimmte Art der Annotation festgelegt ist, so dass also fast jedes linguisti-

sche Konzept mit EXMARaLDA annotiert werden kann. Außerdem ist das Programm plattformübergreifend und relativ intuitiv zu bedienen. Ursprünglich wurde es entwickelt, um Gesprächsaufnahmen zusammen mit Transkriptionen und Annotationen speichern und verarbeiten zu können. Häufig wird EXMARaLDA aber auch ohne Aufnahmen verwendet. Technisch gesehen, können in eingelesenen Korpusdaten einzelne Token oder aber beliebig lange fortlaufende Tokensequenzen mit beliebigen Werten versehen werden. Letzteres bezeichnen wir als Annotation von Tokenspannen oder einfach Spannenannotationen. Als Datenformat erzeugt EXMARaLDA ein eigenes XML-Format, in welchem die eingelesenen Daten und alle erzeugten Annotationsebenen getrennt voneinander, aber natürlich aufeinander bezogen, gespeichert sind (diesen XML-Typ nennt man Standoff-XML, im Gegensatz zu Inline-XML, bei welchem die unannotierten Daten gemeinsam mit den Annotationen gespeichert werden).

Marginalie: "EXMARaLDA annotiert Token oder Spannen von Token"

Alternative Programme zu EXMARaLDA sind ELAN (https://tla.mpi.nl/tools/tla-tools/elan/) und das speziell für die Annotation von Videodateien entwickelte Anvil (http://www.anvil-software.org/). Für die Annotation von schriftlichem Text (ohne mündliche Sprachsignale) werden häufig auch Tabellenkalkulationsprogramme wie das LibreOffice Calc (https://de.libreoffice.org/discover/calc/) oder das gleichnamige Produkt von OpenOffice (http://www.openoffice.org/de/product/calc.html; auslaufend) verwendet, die frei verfügbare Alternativen zu dem von Microsoft entwickelten Excel (https://products.office.com/de-de/excel) sind. Vor jedem Annotationsprojekt ist genau abzuwägen, welches Annotationswerkzeug bzw. welche Annotationswerkzeuge (häufig werden mehrere in einer Verarbeitungskette oder parallel verwendet) hinsichtlich der geplanten Analysen am geeignetsten ist bzw. sind. Aus diesem Grund existiert für Annotationen keine Standardlösung.

Marginalie: "Die Verwendung des geeignetsten Annotationswerkzeugs ist immer zielabhängig"

Die grundlegenden Funktionen von EXMARaLDA für die Annotation schriftlicher Textdaten seien mit Blick auf viele der in den nachfolgenden Kapiteln erläuterten Annotationsarten kurz erläutert (folgen Sie den Punkten der Anleitung, um die Bedienung des Programms am besten nachvollziehen zu können):

Anleitung

- Laden und installieren Sie unter http://exmaralda.org/de/offizielleversion/ die EXMARaLDA-Datei (bzw. »Partitur-Editor«). Starten Sie den »Partitur-Editor« (bei Windows-Systemen liegt dieser im Ordner »FobsJMF« des »EXMARaLDA«-Programmordners).
- Lesen Sie eine tokenisierte Textdatei ein. Verwenden Sie dazu die Funktion »File« > »Import ...«. Wählen Sie in dem sich öffnenden Fenster unten den Datentyp »Plain text file« aus und wählen Sie eine Textdatei in einem Verzeichnis Ihres Computers. Vergleichen Sie zu diesem Arbeitsschritt die Abb. 2.1.
  Verwenden Sie zunächst die Textdatei, die unter https://bit.ly/2Oge2MR als Download angeboten wird. Diese Datei ist ein nach Wörtern und Satzzeichen tokenisierter (segmentierter) Text (Kap. 2.2.5). (Für ein detailliertes Anlegen eines neuen Projekts mit

Ist für einfache Anführungszeichen ← Anm.: In diesen Fällen (Wiedergabe von Programmfehlern) bin ich eher der Meinung, dass sie wie Zitate behandelt werden sollten.

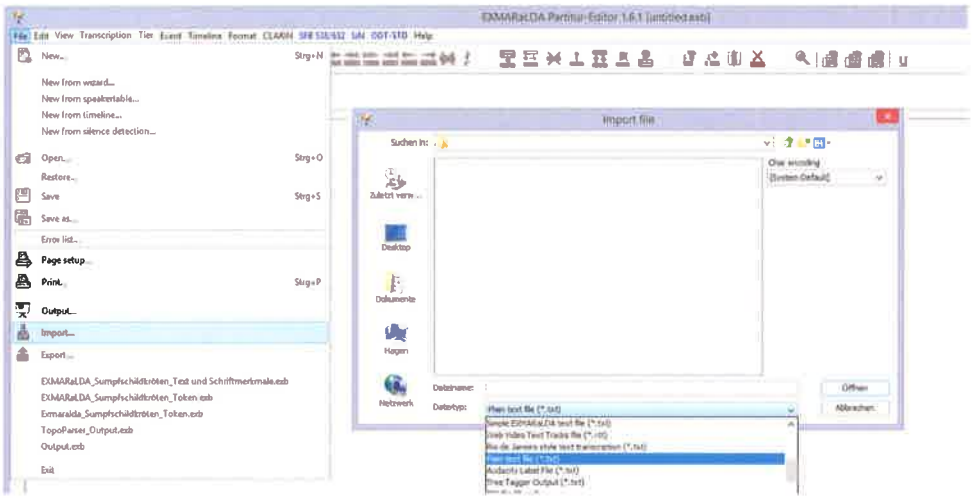


Abb. 2.1: Importfunktion des EXMARaLDA-Partitur-Editors

Audiodaten als Transkriptionsgrundlage vergleichen Sie bitte Kap. 2.4.2.)

- Im nächsten Auswahlschritt wird dem System angegeben, welche Merkmale in der eingelesenen Datei als Grenzsymbole bzw. Separatoren zu interpretieren sind. Dies ist relevant für eine korrekte Zerlegung der eingelesenen Daten in Elementarbestandteile. Wenn Sie die gegebene Testdatei einlesen wollen, ist die oberste Einstellung »Split at paragraphs« korrekt – Sie erhalten in EXMARaLDA einen gemäß den Absätzen segmentierten Text.
- Sollten Sie eine Datei einlesen wollen, in der eine Zerlegung anhand von Leerzeichen erfolgen soll, so wählen Sie die unterste Einstellung »Split at regular expression:« und geben Sie ein Leerzeichen in das Textfeld ein. (Analog hierzu kann jedes andere Zeichen als Separator eingegeben werden. Die Notationsform »(x|y|z)« kann für mehrere Separatoren x, y und z verwendet werden.)
  - Vergleichen Sie z. B. die durch @-Zeichen segmentierte Datei, die unter https://bit.ly/2Fj3mcg beziehbar ist. (Wie eine solche Datei erstellt wird, wird in Kap. 2.2.5 behandelt.) Um diese Datei korrekt zu importieren, wählen Sie »Split at regular expression:« und geben Sie das @-Zeichen als Separator ein. Um nach dem Import dieses Zeichen selber wieder zu eliminieren, wählen Sie in EXMARaLDA die Option »Edit« > »Replace in events...«, geben Sie unter »Search string:« das zu eliminierende @-Zeichen ein und ersetzen Sie mit nichts (das Eingabekästchen unter »Replace string« bleibt leer«). Klicken Sie nun auf »Replace all« und schließen anschließend das Fenster.
- Zum Hinzufügen einer Annotationsebene wählen Sie in der obersten Menüleiste die Funktion »Tier« > »Add tier...« aus. Tiers sind die verschiedenen Transkriptions- Annotations- und Kommentarzeilen bzw. Bearbeitungsebenen im Korpus. Geben Sie im nun erscheinenden Auswahlfenster an, dass Sie eine Annotation hinzufügen wollen, indem

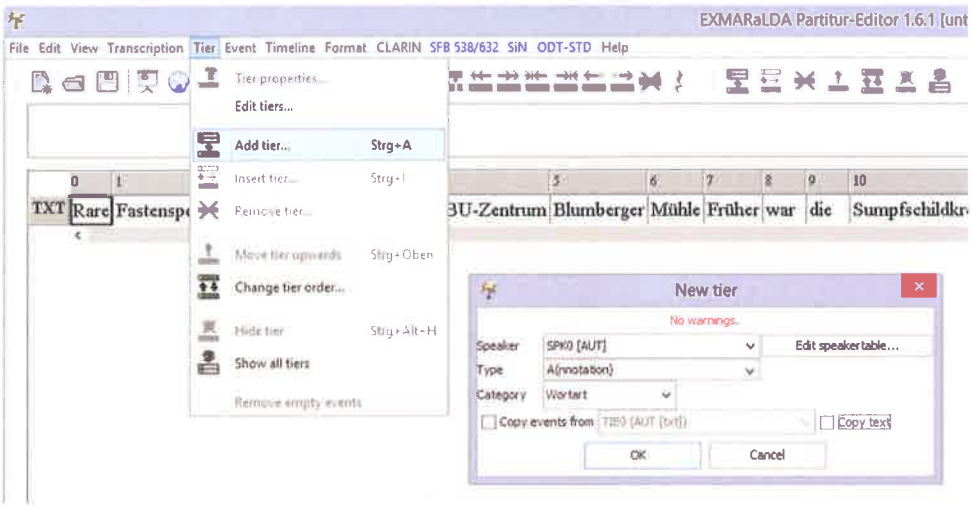


Abb. 2.2: Hinzufügen von Annotationsebenen (Tiers) in EXMARaLDA

Sie unter »Type« die Einstellung »Annotation« auswählen. Geben Sie anschließend einen Namen für die Annotationsebene an. Es entsteht entsprechend eine Zeile, in der die bezeichnete Annotation (z. B. die Wortart) durchgeführt werden kann.

- Die einzelnen Zellen auf jeder Ebene (jedem Tier) können per Mausklick ausgewählt werden und mithilfe der Computertastatur oder den verschiedenen EXMARaLDA-Tastaturen, die unter dem Menüpunkt »View« in der obersten Menüzeile verfügbar sind, editiert werden. Mit der Tab-Taste der Computertastatur springt man zur nächsten Zelle. Man kann außerdem Zellen zusammenfügen, trennen (einfach oder doppelt splitten), Zelleninhalte nach links oder rechts verschieben oder Inhalte aus Zellen löschen. Hierzu dienen die mit der Maus anklickbaren Symbole im links-oberen Bildschirmbereich (zum Zusammenfügen muss man mit der Maus oder dem Touchpad mehrere aneinandergrenzende Zellen markiert haben; zum Splitten muss sich der Cursor an einer bestimmten Stelle innerhalb einer Zelle befinden).

Arbeitsaufgabe

- Lesen Sie anhand der vorangegangenen Informationen die nach Teilsätzen tokenisierte Datei (Arbeitsaufgabe in Kap. 2.2.5; https://bit.ly/2Om4RdS) in EXMARaLDA ein, so dass die Tokensegmentierung korrekt wiedergegeben wird und die Separatoren in der Ergebnisdatei gelöscht sind.
- Speichern Sie das Ergebnis als EXMARaLDA-Datei namens »EXMARaLDA_Import_tokenisiert_3.exb« ab.

2.2.3 | Annotation von Schrift- und Textmerkmalen

Bevor wir zu den Standardverfahren der Erstellung von Annotationen kommen, soll diskutiert werden, wie Texteigenschaften, die durch Standardverarbeitungsverfahren herausnormalisiert werden, als Annotationen gespeichert werden und somit zum Zweck linguistischer Untersuchungen bewahrt werden können.

Schrift- und Textmerkmale: Nehmen wir an, der folgende Textausschnitt einer Webseite soll korpuslinguistisch aufbereitet werden.

Rare Fastenspeise

Sumpfschildkröten im NABU-Zentrum Blumberger Mühle

Früher war die Sumpfschildkröte in Mitteleuropa weit verbreitet. Als Wasserbewohner galt sie frommen Katholiken nicht als Fleisch, sondern als Fisch. Schildkröten durften deshalb auch in der Fastenzeit gegessen werden.

Abb. 2.3: Screenshot eines auf http://www.nabu.de veröffentlichten Artikels (Quelle: https://bit.ly/2OmG4q0)

An dem gegebenen Beispiel kann die Auswirkung automatischer korpuslinguistischer Verfahren auf textuelle Merkmale gut illustriert werden: Wendet man bspw. das Vorverarbeitungswerkzeug des OPenNLP-Toolkits (Tokenisierer; https://opennlp.apache.org/) an, das auch auf der Webseite zur Verarbeitung von Korpusdaten »WebLicht« (https://weblicht.sfs.uni-tuebingen.de; s. Kap. 2.3) verfügbar ist, so erhält man den folgenden segmentierten Text (die einzelnen Segmente sind hier durch Pipes getrennt, um die Segmentgrenzen eindeutig anzuzeigen):

```
Rare|Fastenspeise|Sumpfschildkröten|im|NABU-Zentrum|Blum-
berger| Mühle|Früher|war|die|Sumpfschildkröte|in|Mittel-
europa|weit|verbreitet|.|Als|Wasserbewohner|galt|sie|from-
men|Katholiken|nicht|als|Fleisch|,|sondern|als|Fisch|.|
Schildkröten|durften|deshalb|auch|in|der|Fastenzeit|
gegessen|werden|.
```

Diese Segmentierung (Tokenisierung, s. Kap. 2.2.5) ist nach der Maßgabe, dass jedes Wort- und Satzzeichen genau eine Segmenteinheit sein soll, einwandfrei, und mit solch einer Vorverarbeitung lässt sich ideal weiterarbeiten. Doch sämtliche in Abb. 2.3 ersichtlichen Merkmale der Textgestaltung sind verloren gegangen. Diese sind konkret: Schriftfarbe, Schriftgröße, Absätze, Zugehörigkeit der Wörter zu Überschrift, Unterüberschrift und Artikeltext. Weitere solche Schrift- und Textmerkmale können sein: Unterstreichungen, Fettmarkierungen, Kursivschrift, Seitenumbrüche, Textausrichtung, Hyperlink, Textzusätze und -kommentare (z. B. Randkommentare).

Bewahrung textueller Merkmale: Sofern solche Merkmale als relevant für die spätere Korpusauswertung erachtet werden, müssen Wege gefunden werden, diese Informationen zu bewahren.

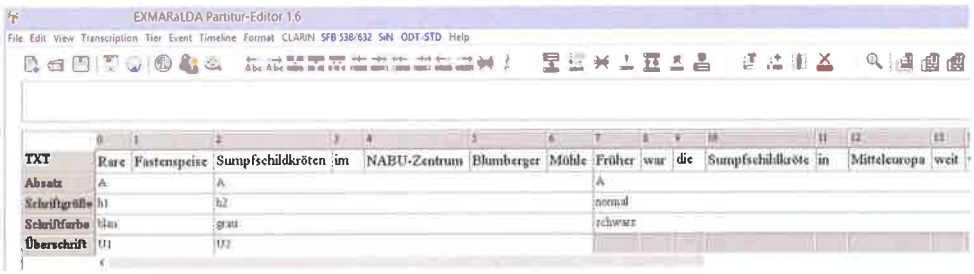


Abb. 2.4: Annotation der in Abb. 2.3 sichtbaren Merkmale »Absatz«, »Schriftgröße« und »Schriftfarbe«

Wenn man weiß, dass gewisse Schrift- und Textmerkmalen von korpuslinguistischen Programmen nicht verarbeitet werden können, muss man sie den Daten als Annotationen hinzufügen. Bezogen auf das oben dargestellte Problem der in Abb. 2.3 vorhandenen Schrift- und Texteigenschaften könnten die in Abb. 2.4 dargestellten Annotationen eine Lösung sein (die Annotationen wurden mit EXMARaLDA erstellt).

In Abb. 2.3 sehen Sie den Text, anhand dessen im vorausgegangenen Kapitel 2.2.2 die Import- und wesentlichen Verarbeitungsfunktionen des EXMARaLDA-Partitureditors erläutert wurden. Dieser Text wurde auf vier Annotationsebenen (Tiers) linguistisch beschrieben: Die oberste Ebene »Absatz« bildet durch entsprechende Spannen unter den fortlaufenden Texttoken die Längen der einzelnen Absätze ab. Als Annotationswert wurde jeweils ein »A« verwendet (leere Spannen oder andere Bezeichnungen sind genauso denkbar). Die Ebene darunter bildet die jeweils verwendete Schriftgröße ab. Als Annotationswerte (Tagset) wurden »h1« für »große Überschrift«, »h2« für »kleine Überschrift« und »normal« für »normaler Fließtext« verwendet (andere Abstufungen und Bezeichnungen sind je nach Textgrundlage beliebig denkbar). Wiederum darunter wird die jeweilige Schriftfarbe in Form der entsprechenden Farbadjektive spezifiziert (alternative Werte könnten auch HTML-Codes für Schriftfarben o. Ä. sein). Zuletzt werden Überschriften in einem zweistufigen System annotiert, indem der Wert »Ü1« für die Hauptüberschrift bzw. den Titel und der Wert »Ü2« für die Unterüberschrift bzw. den Untertitel verwendet wird. Der eigentliche Zeitungstext wird nicht gekennzeichnet, was aber natürlich möglich wäre.

Arbeitsaufgabe

Bilden Sie die in Abb. 2.4 gezeigten Annotationen für die Schrift- und Textmerkmale des NABU-Artikels auf der Webseite https://bit.ly/2OmG4q0 nach (s. Abb. 2.3), damit Sie sich mit der manuellen Annotation im EXMARaLDA-Partitureditor vertraut machen.

2.2.4 | Normalisieren

Definition

Eine **Normalisierung** bedeutet bei der Korpuserstellung, dass ein bestimmtes Merkmal, das in einer Datenmenge heterogen ausgeprägt ist, angeglichen wird.

Normalisieren von Daten findet in vielerlei technischen Kontexten statt. So bedeutet das Normalisieren von Tonaufnahmen, dass die Lautstärke auf ein einheitliches Niveau angeglichen wird. In diesem Sinne verwenden wir hier eine ganz allgemeine Definition von Normalisierung: Bei der Erstellung von Korpora bedeutet ›Normalisierung‹ die Angleichung verschiedener linguistischer Merkmale, z. B.:

- **Schrifttyp, Schriftgröße und andere die Textformatierung betreffende Merkmale:** Es werden ein einheitlicher Schriftstil und in denselben Kontexten dieselben Zeichen verwendet. So können z. B. verschiedene Arten der Anführungszeichen zusammengefasst werden.
  Dies geschieht bei der Verarbeitung der Textdaten meistens automatisch bzw. als gewünschter oder ungewünschter Nebeneffekt.
- **Orthographie:** Heterogene Schreibungen werden auf jeweils eine Form vereinheitlicht. Dies kann sich auf Schreibvarianten beziehen, die während einer gültigen Schreibnorm freigestellt sind (*aufgrund – auf Grund*), aber auch auf historisch diverse Formen (*behände – behende*). Auch die Fehlerkorrektur ist eine Form der Normalisierung, weil hier normgerechte und nicht normgerechte Schreibungen [und in anderen Fehlerbereichen andere Ausdrücke] vereinheitlicht werden. Bei Korpora, die das Gesprochene abbilden, werden häufig zunächst lautgetreue Schreibungen in einem Normalisierungsschritt der orthographischen Norm angepasst.
- **Flexion:** Durch das Zuweisen von Lemmata (Grundformen) werden verschiedene Flexionsformen auf dasselbe Wort bezogen (s. Kap. 2.2.6.3). Auch diesen Vorgang kann man als Normalisierungsprozess bezeichnen, sofern man, wie weiter oben geschehen, Normalisierung abstrakt als Angleichungsvorgang [für variable Einheiten] definiert.
- **Lexik:** Man kann auch verschiedene Lexeme mit derselben Bedeutung zusammenführen, wie es z. B. in historischen Korpora in Form von Hyperlemmata Praxis ist (vgl. z. B. Dipper et al. 2004, S. 24).

Normalisierung vs. Annotation: Häufig werden die Normalisierung und Annotation von Korpusdaten als zwei verschiedene Verarbeitungsschritte angesehen. Methodisch korrekter ist aber die Auffassung, dass die Normalisierung eine (wenn auch sehr spezifische) Art der Annotation ist: Beim Normalisieren interpretiert man die zugrunde liegenden Daten nach bestimmten Richtlinien der Homogenisierung. Dies bedeutet niemals, dass Originaldaten bzw. die zugrunde liegenden Daten überschrieben und somit verfremdet und gelöscht werden müssen. In modernen Korpora werden Normalisierungsebenen nämlich immer als zusätzliche Ebe-

nen eingefügt, ohne dass die Ebene, auf die sich die Normalisierung bezieht, gelöscht wird.

Das Ziel einer korpusbezogenen Normalisierung ist immer, dass die Nutzerinnen und Nutzer alle Instanzen einer abstrakten Form finden können: Wenn Sie alle Vorkommen des Verbs haben finden wollen, möchten Sie in der Korpussuche geeigneterweise mit der Form ›haben‹ alle Vorkommen des Verbs *haben* finden. Dies funktioniert aber nur, wenn Sie auf eine Ebene zugreifen können, in der die Flexionsvarianten *habe*, *hast*, *hat* usw. mit genau dieser Annotation versehen werden. Ggf. sollten auch lautliche Varianten in wörtlicher Rede wie *habm*, *ham* usw. und im Idealfall fehlerhafte Schreibungen wie *habne* auf die Grundform *haben* bezogen werden.

Die Art der Normalisierung und die grundlegende Entscheidung, ob überhaupt in einem eigenen Verarbeitungsschritt normalisiert wird, hängt von den Korpusdaten ab. Sollten Sie stark standardisierte Textdaten wie journalistische Texte verarbeiten wollen, besitzen Sie im Grunde genommen normalisierte Daten und können auf diesen Schritt verzichten. Wenn Sie gesprochene Sprache oder konzeptionell mündliche Sprache (freizeitbasierte Internetforen usw.) verarbeiten wollen, werden Sie normalisieren müssen, sonst können die Korpusnutzer nicht antizipieren, mit welchen Suchausdrücken sie bestimmte Wörter und Konstruktionen finden, und weitere Verarbeitungsschritte werden schwierig. Gehen Sie also davon aus, dass wir zur Anreicherung von Sprachdaten mit ›echten‹ linguistischen Interpretationen einen normalisierten Text benötigen. Häufige weitere Verarbeitungsschritte (Annotationen) werden in den nachfolgenden Kapiteln behandelt.

Konkrete Normalisierungsbeispiele und automatische Analysewerkzeuge zum Normalisieren von schriftlichen Korpusdaten werden in Kapitel 2.4.10 vorgestellt.

Arbeitsaufgabe

- Lesen Sie die Datei, die unter der Webadresse https://bit.ly/2FrgXPY verfügbar ist, in EXMARaLDA ein.
  - Nutzen Sie dazu die »File«-»Import...«-Funktion und wählen Sie das Datenformat »Plain text file (*.txt)«.
  - Geben Sie anschließend unter der Option »Split at regular expression:« ein Leerzeichen ein.
    (Die importierten Daten entsprechen der unter der Webadresse https://bit.ly/2CuMqik verfügbaren Datei. Diese kann alternativ über die Funktion »File« > »Open« im Partitureditor oder über das Ausführen der Datei mit dem Programm EXMARaLDA bzw. »PartiturEditor*.exe« bearbeitet werden.)
- Hierbei handelt es sich um einen Text mit gewissen orthographischen Fehlern wie kleingeschriebenen Nomina. Werden diese nicht normgerechten Formen im Korpus ohne Normalisierung weiterverarbeitet,

können sie später schwer aufgefunden werden und bergen ein hohes Risiko, bei automatischen Verarbeitungsverfahren falsch kategorisiert zu werden. Dies heißt ausdrücklich nicht, dass solche Normverstöße nicht selber von linguistischem Interesse sein können und gänzlich aus den Korpusdaten eliminiert werden sollen. Deshalb ist es wichtig, im Korpus viele unterschiedliche Beschreibungsebenen zu haben, die in ihrer Summe möglichst viele Eigenschaften der im Korpus verarbeiteten Primärdaten abbilden.

  - Fügen Sie in diesem Sinne eine Annotationsebene »Norm« hinzu (»Insert tier«), übernehmen Sie bei der Erzeugung dieser Ebene die Elemente der Ebene »TXT« (»Copy events ...«) und bearbeiten Sie die Annotationsebene »Norm«, ohne die Text-Ebene (»TXT«) zu verändern, so dass auf der Beschreibungsebene »Norm« ausschließlich Formen nach der Standardschreibung (so, wie man die Formen im Lexikon suchen würde) stehen. Es kann nicht nur sein, dass Sie hierfür den Inhalt in den Zellen verändern müssen, sondern auch, dass Sie ggf. Zellen auftrennen bzw. Zellen hinzufügen (Funktion: »Split«) oder Zellen zusammenführen müssen (Funktion: »Merge«).
    Achten Sie jedoch darauf, dass am rechten Rand einer jeden Zelle das Leerzeichen erhalten bleibt. Dann können Sie anschließend die gesamte Spur »Norm« markieren, kopieren, und den Inhalt in eine Textdatei hineinkopieren (ohne Leerzeichen erhalten Sie hierbei eine fortlaufende Zeichenkette).
- Speichern Sie das Ergebnis unter dem Namen »Normalisieren_normalisiert.exb«. Speichern Sie auch eine Textdatei »Normalisieren_normalisiert.txt«, indem Sie den Inhalt der »Norm«-Spur in eine Textdatei kopieren und diese speichern.

2.2.5 | Tokenisieren

Definition

Ein **Token** ist die kleinste zählbare Einheit im Korpus. In einem Korpus wird die Definition der Token spezifisch festgelegt und gilt immer für alle Daten im Korpus.
Subtoken sind alle zu einem Token gehörenden Elemente. Wenn z. B. einzelne Wörter im Korpus als Token definiert werden, so sind die zu den Wörtern gehörenden Zeichen Subtoken.
Tokenisieren von Korpusdaten bedeutet, die zu verarbeitenden Daten in kleinste zählbare Einheiten (Token) zu zerlegen, wobei diese kleinste Einheit konzeptionell nicht festgelegt ist.

Tokenisierung vs. Annotation: Wer die Definition der Annotation noch im Hinterkopf hat, wird feststellen, dass bereits das Tokenisieren eine bestimmte Art der Annotation ist, denn durch den Prozess des Tokenisierens werden die eigentlichen Primärdaten (z. B. ein standardorthographi-

scher Zeitungstext) verändert und es wird bereits beim Tokenisieren eine bestimmte linguistische Information, z. B. die Information über Wortgrenzen, zugewiesen. Dies ist absolut korrekt, auch wenn häufig Tokenisieren und Annotieren als unterschiedliche Verarbeitungsschritte behandelt werden.

Verschiedene Tokendefinitionen: Man kann einen Text tokenisieren, indem man ihn in Absätze, Sätze, Teilsätze, Wörter, Morpheme, Zeichen, Laute usw. zerlegt. Der einzige Anspruch der Tokenisierung ist, dass sie einheitlich erfolgt. Fundamental wichtig ist, dass nach der Verarbeitung des Korpus nach einer bestimmten Maßgabe zur Tokenisierung sämtliche Subtoken – die Elemente also, die kleiner als ein Token sind (z. B. Wörter, wenn Sie satzweise tokenisiert haben) – bei der Korpusauswertung schwer bis gar nicht berücksichtigt werden können. Für die meisten Anwendungsfälle der Korpusaufbereitung ergibt sich, dass jedes Wort und Satzzeichen als Token gilt. Die Gründe hierfür sind,

- dass dies eine sehr intuitive und relativ einfach umzusetzende Richtlinie ist,
- dass schriftliche Texte, die der Korpuserstellung als Rohmaterial zugrunde liegen, durch ihre Spatiensetzung bereits zu einem hohen Grad vorsegmentiert sind (die Ausnahmen und Problemfälle behandeln wir unten),
- dass für die meisten Fragestellungen, die wir an Korpora richten, das Wort eine geeignete Bezugsgröße ist und
- dass sämtliche weitere automatische Verarbeitungsschritte wie das syntaktische Parsing und die Eigennamenerkennung als Eingabe einen nach Wörtern tokenisierten Text erwarten (oder eben diesen Verarbeitungsschritt selber durchführen).

Als Segmentgrenzen (Tokenisierungsseparatoren) zwischen den Token können Sie jedes beliebige Element wählen, das in den Textdaten sonst nicht verwendet wird. Betrachten Sie den folgenden Fall: Sie könnten den Bindestrich als Trennzeichen für Token definieren, müssten dann aber bei jedem Bindestrichkompositum kenntlich machen, dass der Bindestrich hier keine Tokengrenze, sondern ein tokeninternes Zeichen (ein Subtoken) darstellt.

Standardszenario für das Tokenisieren: Normalerweise werden Wörter und Satzzeichen als Token segmentiert und Leerzeichen markieren die Grenze zwischen den Token. Hieraus ergibt sich notwendigerweise, dass die in einem Standardtext nicht von Wortformen abgetrennten Satzzeichen zusätzlich abgetrennt werden müssen, da sie sonst als zu den Wortformen zugehörig interpretiert werden würden.

Tokenisierungsbeispiel

Untokenisierter Text:

```
Ich schreibe dir morgen einen ausführlichen Brief. Da
kann ich dir alles am besten erklären.
```

Tokenisierter Text (mit Leerzeichen als Token-Separatoren):

```
Ich schreibe dir morgen einen ausführlichen Brief . Da
kann ich dir alles am besten erklären .
```

Nach dieser Maßgabe kann das Tokenisieren vor allem in zwei Fällen zu Problemen führen, die der linguistischen Analyse bedürfen, und zwar,

1. wenn ein Leerzeichen im Text keine Wortgrenze markiert, wie z. B. bei Ortsnamen (*Rothenburg ob der Tauber*) und entlehnten Wörtern, insbesondere Namen mit Spatienschreibung (*Rock 'n Roll*) oder
2. wenn Zeichen, die normalerweise als Satzbeendungszeichen fungieren, wortintern auftreten (*usw., z. B., 10.000*).

Am Ende dieses Abschnitts befindet sich eine Aufgabe zum Tokenisieren nicht tokenisierter Textdaten. Die Tokenisierungsaufgabe dient alleine zum Verständnis des Prozesses, der beim Tokenisieren stattfindet: Die Strukturen im Korpus werden nach bestimmten Kriterien in minimale Segmenteinheiten zerlegt. Diese relativ mechanische Arbeit (auch wenn sie immer auch echter linguistischer Entscheidungen bedarf) wird in der Aufbereitungspraxis nicht manuell, sondern automatisch durchgeführt, und zwar entweder in einem gesonderten Verarbeitungsschritt durch ein Tokenisierungsscript oder im Rahmen eines komplexeren Verarbeitungsschritts, z. B. mitsamt der automatischen Annotation von Wortarten und Lemmata (s. Kap. 2.2.6.1).

Automatische Werkzeuge zum Tokenisieren: Meistens wird das automatische Tokenisieren von Korpusdaten im Rahmen weiterer Verarbeitungsschritte durchgeführt, so dass kein isolierter Tokenisierungsvorgang nötig ist. Folgende drei Möglichkeiten zum Tokenisieren nicht tokenisierter Textdaten sollen jedoch kurz erwähnt werden:

1. Nutzen Sie den TreeTagger (https://bit.ly/1bsE7eE), wie in Kapitel 2.2.6.1 beschrieben, und behalten Sie in der dreispaltigen Ausgabedatei lediglich die linke Spalte (dies ist der tokenisierte Text).
2. Laden und entpacken Sie das TreeTagger-Paket (https://bit.ly/1bsE7eE), das das Perl-Script »tokenize.pl« enthält. Rufen Sie per Kommandozeile im Verzeichnis, in welchem das Script liegt, den Befehl
   ```
   tokenize.pl [Name der zu tokenisierenden Datei] >[Name der auszugebenden Datei]
   ```
   auf. Auf Ihrem Computer muss Perl installiert sein (http://www.perl.org/get.html).
3. Nutzen Sie in dem Webdienst WebLicht (https://weblicht.sfs.uni-tuebingen.de/; s. Kap. 2.3) den »Advanced Mode« und tokenisieren Sie eine eingelesene Datei oder eingelesenen Text mithilfe des OpenNLP-Tokenisierers (Teil der OpenNLP-Toolchain, https://opennlp.apache.org/).

Arbeitsaufgabe

- Beziehen Sie von der Internetadresse https://bit.ly/2CxEr3R einen untokenisierten Text. Hierbei handelt es sich um den bereits normalisierten Text aus der Normalisierungsaufgabe in Kap. 2.2.4.
- Tokenisieren Sie den Text innerhalb der Datei nach den folgenden Maßgaben:

Nr.	Token-Definition	Tokenisierungsseparator
1.	Ein Token ist genau ein Wort oder ein Satzzeichen.	Die Tokengrenze wird durch Absätze angezeigt. Leerzeichen sind nur bei wortinternem Gebrauch legitim.
2.	Ein Token ist genau ein Teilsatz.	Die Tokengrenze wird mit Rautenzeichen (»#«) angezeigt. Leerzeichen markieren Wortgrenzen und gehören somit zu den einzelnen Token (sie sind Subtoken).
3.	Ein Token ist genau ein Morphem oder ein Satzzeichen.	Die Tokengrenze wird durch @-Zeichen markiert. Leerzeichen sind kein legitimes Zeichen im Korpus.

- Speichern Sie die jeweils bearbeitete Datei gesondert mit dem Zusatz »_tokenisiert_1« bzw. »_tokenisiert_2« und »_tokenisiert_3«.

2.2.6 | Typische tokenbasierte Annotationen

In den folgenden Unterkapiteln werden häufig verwendete und meistens automatisch erzeugte Annotationen vorgestellt, die sich in aller Regel auf die fortlaufenden Wortformen im Korpus beziehen, also tokenbasiert sind. Hierzu sei kritisch angemerkt, dass zum einen Token nicht zwingend wortbasiert sind und dass praktisch jedes linguistische Konzept auf verschiedene Weisen darstellbar ist. Die Annahme, die nachfolgenden Annotationen seien Tokenannotationen, ist demnach stark verallgemeinert und muss im Einzelfall nicht der Tatsache entsprechen.

2.2.6.1 | Tagging: Wortarten und Lemmata

›**Tagging**‹ bzw. das aus dem englischen abgeleitete Verb ***taggen*** bezeichnet das Annotieren von tokenisierten Daten mit Werten für Wortarten und Grundformen (Lemmata). Da *tag* im Englischen ganz allgemein »kennzeichnen, markieren« (bzw. »Kennzeichnung«) bedeutet, kann man den Begriff ›Tagging‹ allerdings auch auf andere Annotationen beziehen (es gibt somit eine enge und eine weite Auffassung des Begriffs).

›Tagging‹ umfasst also in seiner engeren Bedeutung genau zwei Annotationsschritte, das Annotieren von Wortarten (auch ›pos-Tagging‹ genannt, weil die Abkürzung der englischen Bezeichnung für Wortarten *part of speech* ›pos‹ ist) und das Lemmatisieren.

Gehen wir von einer wort- und satzzeichenbezogenen Tokenisierung der Korpusdaten aus, so ist das Ziel, dass jedes Token genau einen Wert für eine Wortart (und beim Lemmatisieren genau eine Grundform) erhält.

2.2.6.2 | Wortartenanalyse im Taggingprozess

Wenn Sie ein Linguistikstudium absolvieren oder absolviert haben, werden Sie, zumindest was die Analyse von Wortarten angeht, gelernt haben, dass diese Aufgabe alles andere als trivial ist. Dies liegt in erster Linie daran, dass Wörter ein sehr komplexes Gefüge von Eigenschaften auf verschiedensten grammatischen Ebenen besitzen und dass man sich in der Tradition der Wortartenanalyse nicht auf die ausschlaggebenden Eigenschaften bzw. eine bestimmte Art der Priorisierung der verschiedenen Eigenschaften geeinigt hat. Man muss deshalb beim Tagging unbedingt im Hinterkopf behalten, dass sämtliche Probleme, die in der deskriptiven Grammatik bei der Beschreibung von Wortarten auftreten, auch im Kontext der korpusbasierten Wortartenzuweisung relevant sind. Die grundlegendsten Probleme seien hier als (kommentierte) Fragen aufgelistet:

- Wie sieht der Bestand an Wortarten einer Sprache aus; wie viele Wortarten gibt es? Sie wissen, dass verschiedene Beschreibungstraditionen und Theorien hierauf verschiedene Antwort geben – die Anzahl der weniger ausdifferenzierten Wortarten (ohne semantische oder flexionsmorphologische Unterklassen) variiert von ca. 7–8 bis ca. 15. Dies hängt mit der nächsten Frage zusammen:
- Welche Typen von Eigenschaften sollen hauptsächlich als Unterscheidungskriterien verwendet werden? Als grundlegende Ebenen der Beschreibung von Wortarten dienen die Syntax (z. B. die Position eines Wortes im Satz oder einer Wortgruppe sowie seine Rektionseigenschaften – eine Präposition ist ein vorangestelltes, kasusregierendes Wort), die Flexionsmorphologie (z. B. die Beugbarkeit oder Gebeugtheit eines Wortes – ein Infinitum ist ein nicht gebeugtes Verb), die Wortbildungsmorphologie (z. B. können bestimmte Wortbestandteile zur Kategorisierung führen – Pronominaladverbien werden häufig als *da(r)*-Wörter mit präpositionalem Zweitbestandteil definiert), bisweilen entscheidet aber auch die Semantik (ein Demonstrativum ist zeigendes Wort), die Pragmatik (eine Gesprächspartikel ist spezifisch für den Verwendungskontext Gespräch) oder die Informationsstruktur (eine Anadeixis ist ein in den Präkontext zeigendes Wort) über die Kategorisierung.
- Wie lexikalisch festgelegt ist die einzelne Wortform bzw. wie ausschlaggebend ist der Kontext für die Kategorisierung? Ist ein Adjektiv wie *gut* (*Das ist gut.*) noch ein Adjektiv, wenn es adverbial verwendet wird (*Peter spielt gut.*)?

Das STTS als Standard-Tagset für die Wortartenanalyse im Deutschen: Wie bereits erwähnt, besteht bezüglich dieser Fragen kein Konsens und erfahrene Linguistinnen und Linguisten werden mit Recht fordern, dass diese Fragen je nach Analyseziel unterschiedlich zu beantworten sind. Die Auflistung und Zuweisung von Wortarten ist somit relativ arbiträr. Im Kontext der korpusbasierten Sprachbeschreibung hat sich dennoch ein bestimmtes Beschreibungsinventar (ein Tagset) an Wortarten als Standard durchgesetzt: das STTS-Tagset. Eine Liste der im STTS vorgesehenen Kategorien (Tags) finden Sie unter der Webadresse https://bit.ly/1KWyVVM. Ausführliche Anweisungen zur Vergabe dieser Tags finden Sie in den Richtlinien Schiller et al. 1999: https://bit.ly/2TVWURK. Dieses Tagset sieht 50 verschiedene Tags zur Vergabe von Wortartkürzeln (und Satzzeichenkürzeln) vor. Die Zahl der Werte ist deshalb so hoch, weil das Tagset hierarchisch aufgebaut ist und zusätzlich zu grundlegenden Kategorien wie Verb oder Pronomen auch spezifische Informationen zum Beugungsstatus, der syntaktischen Verwendung oder der Bedeutung abgefragt werden.

So werden Verben (V) z. B. in die Untertypen Vollverb (VV), Hilfsverb (VA) und Modalverben (VM) eingeteilt, in denen jeweils noch die Flexionsstatus finit (VVFIN, VAFIN, VMFIN), infinit (VVINF usw.), partizipial (VVPP usw.) sowie imperativisch (VVIMP usw.) unterschieden werden.

Beispiel für eine pos-Analyse nach dem STTS: Die Anwendung des STTS-Tagsets auf den Beispielsatz

(1) *Ich schreibe dir morgen einen ausführlichen Brief.*

kann wie folgt dargestellt werden (die STTS-Wortart-Tags befinden sich jeweils mit Schrägstrich abgetrennt hinter dem jeweiligen Token; man nennt dies eine ›Inline-Annotation‹, weil die Ausgangsdaten und die Annotation in einer Zeile bzw. Ebene stehen):

(2) *Ich*/PPER *schreibe*/VVFIN *dir*/PPER *morgen*/ADV *einen*/ART *ausführlichen*/ADJA *Brief*/NN ./$.

Dabei stehen die einzelnen Werte für folgende Kategorien: PPER = Personalpronomen; VVFIN = finites Vollverb; ADV = Adverb; ART = Artikel; ADJA = attributives Adjektiv; NN = normales Nomen; $. = satzbeendende Interpunktion. Eine etwas andere Darstellung der Annotation, in der der tokenisierte Text und die Wortartenanalyse eindeutiger voneinander getrennt sind, ist eine tabellenförmige:

(3)

Ich	PPER
schreibe	VVFIN
dir	PPER
morgen	ADV
einen	ART
ausführlichen	ADJA
Brief	NN
.	$.

In diesem Format sind die zu analysierenden Token durch Tabulatorabstände (Tabstopp, Trennzeichen) von den Annotationen getrennt.

Der beste Weg, sich mit einem Tagset vertraut zu machen, ist, es mitsamt den dazugehörigen Richtlinien auf Korpusdaten anzuwenden.

Arbeitsaufgabe

Annotieren Sie die nach Wörtern und Satzzeichen mittels Absätzen tokenisierte Datei (Arbeitsaufgabe in Kap. 2.2.5; https://bit.ly/2Oge2MR).

- Verwenden Sie dabei das unter https://bit.ly/1KWyVVM verfügbare Tagset. Eine offline verwendbare Version finden Sie hier: https://bit.ly/2TjSWhq.
- Die Annotationsrichtlinien zum STTS finden Sie hier: https://bit.ly/2TVWURK.
- Vergeben Sie die Annotationskürzel (Tags) des STTS entweder, indem Sie den Token in der Textdatei einen Tabulatorabstand anfügen und dann manuell die STTS-Werte ausschließlich in Großbuchstaben hinzufügen, oder kopieren Sie den Dateiinhalt in eine Arbeitsmappe des Programms LibreOffice Calc oder Microsoft Excel und fügen Sie den Token in der jeweils rechts danebenliegenden Tabellenspalte die passenden STTS-Werte hinzu.
- Speichern Sie die annotierte Datei als Textdatei unter dem Namen Tokenisiert_1_getaggt.txt. Wenn Sie im Tabellenprogramm gearbeitet haben, kopieren Sie die zwei Spalten in eine Textdatei zurück oder speichern Sie das Ergebnis als .csv-Datei mit tabulaturgetrennten Spalten. Sie können anschließend die Dateiendung in .txt umbenennen und die Datei in einem einfachen Texteditor öffnen.

2.2.6.3 | Lemmatisierung im Taggingprozess

Lemmatisieren bezeichnet die Annotation sämtlicher Oberflächenformen, also häufig Token, im Korpus mit Grundformen, sog. Lemmata (Singular: Lemma).

Definition

Für das Lemmatisieren gilt dasselbe grundlegende Problem wie für das Wortartentagging: Wir müssen uns mit einem an sich einfachen Verfahren auseinandersetzen, das zwar an manchen Stellen sehr intuitiv und zweifelsfrei funktioniert, an anderen Stellen aber mit Zweifelsfällen und einer entsprechend tiefen linguistischen Auseinandersetzung verbunden ist. Da ›Lemmatisierung‹ die Zuweisung von Grundform bedeutet, will man bei diesem Prozess also jeder Wortform eine Grundform zuweisen, was wiederum zur Folge hat, dass eine Wortform wiederholt wird, wenn sie bereits eine Grundform ist. Das Lemma kann generell nur bei flektierbaren Wörtern von der Oberflächenform abweichen, mit Ausnahme von Groß- und Kleinschreibungsprinzipien: Die Grundform eines satzinitialen ›*Immer*‹ ist ›*immer*‹. Die Grundform einer jeden Form innerhalb eines Flexionsparadigmas ist arbiträr. Ich kann also der Verbform *läufst* die Grundform des Infinitivs *laufen* zuweisen, kann mich

aber auch für die erste Person Singular *laufe*, den Verbstamm *lauf-* usw. entscheiden, wie wir es aus verschiedenen lexikologischen Traditionen kennen.

Im Deutschen lemmatisiert man normalerweise im verbalen Bereich den Infinitiv, im nominalen Bereich den Nominativ Singular (Maskulinum bei genusvariablen Wortarten) und bei Adjektiven die flexionslose Grundform.

Probleme beim Umsetzen der Lemmatisierung sind durch diese Vorgaben an vielen Stellen vorprogrammiert: Was tun wir bei Nomina wie *Beamter*, das eine starke und eine schwache Endung haben kann (*der Beamte – ein Beamter*)? Was geschieht mit Adjektiven, die nicht endungslos (prädikativ) auftreten, wie *viel-* in *die vielen Menschen*, und das außerdem selten im Singular auftritt? Ähnlich problematisch ist der Determinierer *alle* in *alle Menschen*. Wer große Mengen Sprachdaten per Hand annotiert, wird andauernd auf solche Probleme stoßen. Als Fazit ist es nicht sinnvoll, die angesprochenen Probleme aufzulösen, sondern zu schlussfolgern, dass wir viel linguistischen Sachverstand mitbringen müssen, um gute, konsistente Entscheidungen zu fällen, und dass wir diese Entscheidungen dokumentieren müssen, weil nur die klaren Fälle antizipierbar sind: Sie wissen, dass Sie mit einer Lemmasuche nach »*schön*« Wortformen wie *schönem* finden werden, aber Sie wissen nicht, ob Sie die Form *mochte* mit der Lemmasuche »*möchten*« oder »*mögen*« finden, wenn es Ihnen nicht mitgeteilt wird.

Auch was das Lemmatisieren angeht, sollten Sie sich durch rein manuelle Arbeit an authentischen Korpusdaten ein Gefühl für den Prozess verschaffen, sofern Sie noch keine Erfahrung damit besitzen.

Arbeitsaufgabe

Lemmatisieren Sie die nach Wörtern und Satzzeichen mittels Absätzen tokenisierte Datei (Arbeitsaufgabe in Kap. 2.2.5; https://bit.ly/2Oge2MR).

- Vergeben Sie dabei jedem Token genau eine Grundform. Entspricht die Form des Tokens bereits der Grundform, schreiben Sie diese Form noch einmal hin. Achten Sie auch auf die Groß- und Kleinschreibung, gerade an Satzanfängen.
- Vergeben Sie die Annotationskürzel (Tags) des STTS entweder, indem Sie den Token in der Textdatei einen Tabulatorabstand anfügen und dann manuell die STTS-Werte ausschließlich in Großbuchstaben hinzufügen, oder kopieren Sie den Dateiinhalt in eine Arbeitsmappe des Programms LibreOffice Calc oder Microsoft Excel und fügen Sie den Token in der jeweils rechts danebenliegenden Tabellenspalte die passenden STTS-Werte hinzu.
- Speichern Sie die annotierte Datei als Textdatei unter dem Namen Tokenisiert_1_lemmatisiert.txt. Wenn Sie im Tabellenprogramm gearbeitet haben, kopieren Sie die zwei Spalten in eine Textdatei zurück oder speichern Sie das Ergebnis als .csv-Datei mit tabulaturgetrennten

(Trennzeichen-getrennten) Spalten. Sie können anschließend die Dateiendung in .txt umbenennen und die Datei in einem einfachen Texteditor öffnen.

2.2.6.4 | Annotationswerkzeuge zum Wortartentagging und zur Lemmatisierung

Wie wir bereits beim Tokenisieren exemplarisch gesehen haben, gibt es einige Verarbeitungsschritte, die man zwar durch eine manuelle Durchführung genau verstehen kann (und die man als Nutzerin oder Nutzer deshalb unbedingt einmal manuell durchgeführt haben sollte), die aber bei der Erstellung größerer Korpora nicht manuell durchlaufen werden. Der Aufwand wäre deutlich zu hoch und die automatischen Verfahren sind im Bereich des konventionellen Taggings sehr präzise. Deshalb verwendet man sogenannte Tagger – Programme zur automatischen Analyse von Wortarten und/oder Lemmata.

Der TreeTagger: Am verbreitetsten ist hier Helmut Schmids TreeTagger (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/, Schmid 1994). Da der TreeTagger nicht ganz einfach und systemunabhängig zu installieren ist, empfiehlt sich für die unkomplizierte Nutzung des TreeTaggers ein von Laurence Anthony kreiertes Interface TagAnt (http://www.laurenceanthony.net/software/tagant/, Anthony 2015; zur Bedienung s. u.). Es existieren aber auch Plattformen zur Online-Nutzung von Taggern, wie z. B. WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/, s. Kap. 2.3), wo auch andere Tagger für das Deutsche (und viele andere Sprachen) Anwendung finden und bei Bedarf anschließend die Ergebnisse verglichen werden können. Auf der Webseite https://bit.ly/2MZpn2O kann man außerdem den TreeTagger online anwenden.

Die derzeit einfachste Art, Texte, die im .txt-Format gespeichert sind, auf dem eigenen Computer zu taggen, ist die Nutzung von TagAnt. Gehen Sie wie folgt vor, um das Programm zu nutzen.

Anleitung

- Laden Sie TagAnt in der passenden Version für Ihr Betriebssystem von der Internetseite http://www.laurenceanthony.net/software/tagant/ herunter. Die ausführbare Datei, die heruntergeladen wurde, lässt sich ohne weitere Installation verwenden.
- Öffnen Sie TagAnt (durch Doppelklick auf die Datei).
- Aktivieren Sie die Option »Input Files«. Klicken Sie »Load« und wählen Sie eine auf Ihrem Computer befindliche Textdatei mit der Dateiendung .txt. TagAnt erwartet eine UTF-8-Kodierung, in denen viele Textdateien vorliegen und die man mit Texteditoren wie Notepad + + (https://notepad-plus-plus.org/) festlegen kann.
- Wählen Sie bei deutschsprachigen Dateien bei »Language« die Einstellung »German«.
- Wählen Sie die Ausgabeeinstellung »Vertical«.

- Die Programmoberfläche mit den genannten Taggingeinstellungen sollte wie in Abb. 2.5 aussehen.
- Führen Sie durch aktivieren der Funktion »Start« das Tagging mit den gewählten Einstellungen durch. Es wird eine neue Textdatei im Ordner der zu taggenden Datei mit dem Zusatz _tagged erzeugt. Diese Datei enthält drei Spalten: den tokenisierten Text in der linken, STTS-Werte

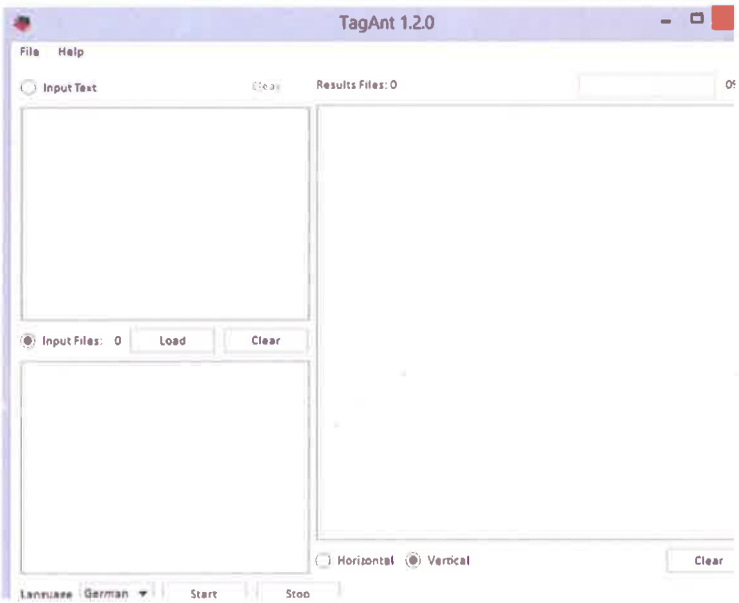


Abb. 2.5: TagAnt mit den genannten Einstellungen ~~und einer geladenen Datei~~ ~~»Normalisiert.txt«~~

für Wortarten in der mittleren und Lemmata in der rechten. Wortartenwerte sowie Lemmawerte werden immer vergeben – Wortartenwerte probabilistisch und Lemmata lexikonbasiert. Ist eine Wortform im Lexikon unbekannt, so wird sie auf der Lemmaebene übernommen.

Weitere Hinweise zur Verwendung von Taggern: Unten in diesem Kapitel finden Sie Aufgaben zum automatischen Tagging von Textdaten. In den Hinweisen zur Aufgabe erfahren Sie, wie man eine manuell erzeugte Analyse mit der automatisch erstellten Analyse des Taggers vergleichen kann. Sie werden sehen, dass Sie zunächst längere Zeit benötigen, um die im STTS-Tagset vorgesehenen Tags manuell zuzuweisen, dass die Geschwindigkeit jedoch rasch zunimmt. Bezogen auf die automatische Analyse kann beobachtet werden, dass der Tagger grundsätzlich eine erstaunlich gute Analyse liefert, dass sich diese jedoch drastisch verschlechtert, wenn der sprachliche Input vom Standard abweicht.

Die Funktionsweise automatischer Tagger lässt sich nicht pauschal beschreiben, weil in der Entwicklung der automatischen Verfahren verschiedene Ansätze konkurrieren, die sich entweder ausschließen oder die in Kombination genutzt werden können. Die ›erfolgreichen‹ Tagger besitzen ein sehr großes Lexikon, in dem Wortformen mit ihren möglichen

Wortarten registriert sind. Durch ein solches Lexikon können Formen, die nicht ambig (mehrdeutig) sind, unmittelbar korrekt getaggt werden. Bei ambigen oder nicht im Lexikon verzeichneten Formen muss ein Verfahren angewendet werden, das zwischen verschiedenen Möglichkeiten der Zuweisung eines Werts genau eine auswählt.

Ganz grundlegend unterscheidet man bei automatischen Analysen regelbasierte und statistische Verfahren. Im Fall einer Form, die anhand des Lexikons nicht eindeutig getaggt werden kann, kann der Wert entsprechend durch eine Regelanweisung oder durch einen Wahrscheinlichkeitswert ermittelt werden. Man kann nicht pauschal sagen, welcher Ansatz der bessere ist, weil je nach Beschaffenheit der zu analysierenden Daten und der Verfügbarkeit von Trainingsdaten ab (Trainingsdaten sind von Hand analysierte Daten, von denen man ausgeht, dass sie korrekt analysiert sind, und die deshalb als Grundlage für einen maschinellen Lernprozess verwendet werden).

Wir können hier nicht weiter in das Problem des maschinellen Lernens und in die Konfiguration der bestehenden Tagger einsteigen. Zusammenfassend seien jedoch wichtige Grundtendenzen für die Abhängigkeit des Taggingergebnisses von den zu analysierenden Sprachdaten genannt:

- Wörter, die besonders selten vorkommen oder ad hoc gebildet sind, so dass sie nicht im Lexikon eines Taggers verzeichnet sind, haben vergleichsweise schlechte Chancen, korrekt getaggt zu werden:
  (1) *Schaufel*/VVIMP *nicht*/PTKNEG *so*/ADV !/$.
  Es ist ziemlich wahrscheinlich, dass auch in einer sehr großen Menge an Trainingsdaten eine großgeschriebene Wortform *Schaufel* nicht als imperativisches Verb vorkommt. Infolgedessen wird ein auf solchen Daten trainierter Tagger *Schaufel* im Lexikon nur mit dem Wert NN verzeichnet haben und somit auch NN zuweisen.
- Nicht normgerecht repräsentierte Wörter, also Fehlschreibungen oder bewusste Normvarianten wie Binnenmajuskelschreibungen usw. finden in der Regel keinen Abgleich mit im Lexikon verzeichneten Formen, womit sich das Taggingergebnis ebenso verschlechtert. Hier kann eine vorab durchgeführte Normalisierung erhebliche Verbesserungen erzielen.

Arbeitsaufgabe

- Taggen Sie die Datei, die zuvor manuell nach Wortarten und Lemmata annotiert werden sollte, mit AntConc. Die Datei können Sie unter https://bit.ly/2CxEr3R beziehen. Verwenden Sie die oben stehenden Handlungsanweisungen für die Verwendung des Taggers.
- Überprüfen Sie das Taggingergebnis, indem Sie die von Tagger ausgegebene Datei Tokenisieren_tagged.txt mit den Ergebnissen des manuellen Wortartentaggings (Tokenisiert_1_getaggt.txt) und des manuellen Lemmatisierens (Tokenisiert_1_lemmatisiert.txt) vergleichen.

Um dies computerunterstützt zu tun, können Sie aus den manuell erstellten Dateien eine Datei mit einer dreispaltigen Tabelle nach dem Vorbild der von TagAnt ausgegebenen Datei erzeugen und anschließend die manuelle und die automatisch erzeugte Analyse mit einem Programm vergleichen: eine passende Funktion besitzt Microsoft Word; ein frei erhältliches Vergleichsprogramm ist KDiff3 (http://kdiff3.sourceforge.net/ bzw. http://sourceforge.net/projects/kdiff3/files/kdiff3/).

2.2.6.5 | Datenrepräsentation im TreeTagger-Outputformat und in alternativen Kodierungen

Wir werden bei der Besprechung der verschiedenen Annotationsmöglichkeiten immer wieder auf die Datenformate zu sprechen kommen, in denen die verschiedenen Annotationen gespeichert werden. Dies ist relevant für das Verständnis der weiteren Verarbeitungswege.

Das TreeTagger-Outputformat: Wenn Sie den TreeTagger anwenden, erhalten Sie oftmals eine dreispaltige Tabelle im Plaintextformat (hier sind die einzelnen Spalten einfach durch Tabulatorabstände voneinander getrennt). Der tokenisierte Text befindet sich in der linken Spalte. Rechts von jedem Token steht in derselben Zeile zunächst die Wortartinformation, dann das Lemma. Für den Satz *Ich schreibe dir morgen einen ausführlichen Brief.* ergibt sich im TreeTagger-Outputformat das in (1) abgebildete Taggingergebnis.

(1)

Ich	PPER	ich
schicke	VVFIN	schicken
dir	PPER	du
morgen	ADV	morgen
einen	ART	ein
ausführlichen	ADJA	ausführlich
Brief	NN	Brief
.		$.

Der Vorteil an einem solchen Speicherformat ist, dass es an keine technischen Anforderungen geknüpft und gut menschenlesbar ist und ohne Probleme weiterverarbeitet werden kann.

Alternative Speicherformate: Mit ein paar Handgriffen lassen sich solche Daten aber auch anders darstellen und speichern. Beachten Sie, dass die folgenden Darstellungen dieselben Informationen enthalten und deshalb verlustfrei ineinander überführt werden können.

Vergleichen Sie (2) als Darstellung des Beispielsatzes in einer einfachen Inline-Annotation, wobei ein Leerzeichen immer eine Tokengrenze und ein @-Zeichen immer einen Annotationsseparator markiert:

(2) *Ich*@PPER@ich *schicke*@VVFIN@schicken *dir*@PPER@du *morgen*@ADV@morgen *einen*@ART@ein *ausführlichen*@ADJA@ausführlich *Brief*@NN@Brief .@$.@.

Vergleichen Sie (3) als Darstellung des Beispielsatzes in XML-kodierter Inline-Annotation:

(3)
```
< tokens >
< word = «Ich« pos = «PPER« lemma = «ich« >
< word = «schreibe« pos = «VVFIN« lemma = «schreiben« >
< word = «dir« pos = «PPER« lemma = «du« >
< word = «morgen« pos = «ADV« lemma = «morgen« >
< word = «einen« pos = «ART« lemma = «ein« >
< word = «ausführlichen« pos = «ADJA« lemma = «ausführlich« >
< word = «Brief« pos = «NN« lemma = «Brief« >
< word = «.« pos = «$.« lemma = «.« >
< /tokens >
```

Diese Anführungszeichen müssen öffnend sein. Bezügl. aller Anführungszeichen in (3) würde ich hochgestellte Anführungszeichen ("ich") bevorzugen.

Arbeitsaufgabe

- Überführen Sie die annotierte Datei, die Sie unter der Webadresse https://bit.ly/2umA0og herunterladen können, in das dreispaltige TreeTagger-Format:
  - Ausgangsformat:
    Lieber@ADJA@lieb Nutzer@NN@Nutzer ...
  - Zielformat:
    Lieber ADJA lieb
    Nutzer NN Nutzer
    ...
- Sie erreichen dies, indem Sie mit einem funktionsreicheren Texteditor (wie Notepad + +, https://notepad-plus-plus.org/) folgende Ersetzungen durchführen:
  - Ersetze @-Zeichen (geben Sie dieses in das »Suchen«-Feld ein) mit Tabulaturabstand (Tabstopp, Trennzeichen; kopieren Sie einen solchen in das »Ersetzen«-Feld).
  - Ersetze Leerzeichen (geben Sie ein Leerzeichen in das »Suchen«-Feld ein) mit Absatz (geben Sie den regulären Ausdruck »\n« für »Absatz« in das »Ersetzen«-Feld und geben Sie an, dass Sie erweiterte Ausdrücke bzw. reguläre Ausdrücke verwenden).
- Das Ergebnis muss aussehen wie die Daten in der getaggten Datei, die Sie mit TagAnt erzeugt haben (s. die Taggingaufgabe in Kap. 2.2.6.4) bzw. die Lösungsdatei, die unter https://bit.ly/2TnaDMY verfügbar ist.

Hier einfache Anführungszeichen

2.2.6.6 | Flexionsmorphologie

Definition

Als **Flexionsmorphologie** wird der Teil der ~~Wortbildung bzw. Wortbildungslehre~~ bezeichnet, der die Flexion (die Wortbeugung) betrifft.

Da im Deutschen viele Kasusformen nicht eindeutig durch Flexionsendungen gekennzeichnet sind, kann der Flexionsstatus nur in wenigen Fällen eindeutig an der Oberflächenform eines gegebenen Wortes abgelesen werden (z. B. ist die Form *dem* eindeutig ein Dativ, aber die Form *einer* kann Nominativ, Genitiv oder Dativ sein). Deshalb ist es in Fällen, in welchen nicht nur die eindeutig markierten Formen ausgewertet werden sollen, notwendig, die Flexionskategorien bei den flektierbaren Wörtern explizit zu kennzeichnen. Hierzu werden, wie wir bereits bei den Wortarten gesehen haben, Kürzel für die Flexionskategorien genutzt, die in aller Regel bei den flektierten Wortformen aneinandergehängt werden. Richtlinien zur Vergabe von Flexionstags in Korpora deutscher Sprache sind in Crysmann et al. (2005; https://bit.ly/2Hwj9rL) ausgearbeitet. Viele manuell annotierte Korpora und Tagger mit flexionsmorphologischer Ausgabe (s. u.) haben ihr eigenes Format der Anordnung und Abkürzung der Flexionskategorien. Da die Werte jedoch relativ selbsterklärend sind (für »Dativ« wird meistens »Dat« verwendet usw.), ist dies nicht besonders problematisch.

Beispiel für eine flexionsmorphologische Analyse nach Crysmann et al. (2005): Die Token im Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* erhalten gemäß den Richtlinien in Crysmann et al. (2005) folgende flexionsmorphologische Werte:

(1) *Ich*/1.Nom.Sg.* *schreibe*/1.Sg.Pres.Ind *dir*/2.Dat.Sg.* *morgen*/- *einen*/Acc.Sg.Masc *ausführlichen*/Pos.Acc.Sg.Masc *Brief*/Acc.Sg.Masc ./-

Ausformuliert bedeuten die Werte jeweils:

(2)

Ich:	Erste Person Nominativ Singular, nicht bestimmbares Genus
schreibe:	Erste Person Singular Präsens Indikativ
dir:	Zweite Person Dativ Singular, nicht bestimmbares Genus
morgen:	Keine Flexionskategorien bestimmbar
einen:	Akkusativ Singular Maskulinum
ausführlichen:	Positiv, Akkusativ Singular Maskulinum
Brief:	Akkusativ Singular Maskulinum
.:	Keine Flexionskategorien bestimmbar

Man sieht, dass einige Flexionskategorien aus dem Kontext ermittelt werden müssen und in manchen Kontexten ~~nicht~~ bestimmbar sind. Außer-

	Ich	*schreibe*	*dir*	*morgen*	*einen*	*ausführlichen*	*Brief*	*MV*
Person	1.	1.	2.	–	–	–	–	–
Kasus	Nom	–	Dat	–	Acc	Acc	Acc	–
Numerus	Sg	Sg	Sg	–	Sg	Sg	Sg	–
Genus	*	–	*	–	Masc	Masc	Masc	–
Tempus	–	Pres	–	–	–	–	–	–
Modus	–	Ind	–	–	–	–	–	–
Steigerung	–	–	–	–	–	Pos	–	–

Tab. 2.1: Abbildung der flexionsmorphologischen Merkmale aus Beispiel (1) und (2) als Mehrebenenannotation

dem handelt es sich im Falle des Merkmals ›Positiv‹ bei den Adjektiv *ausführlichen* streng genommen um kein flexionsmorphologisches Merkmal, sondern um ein wortbildungsmorphologisches. Dies stellt jedoch kein Problem dar, weil es bei der Beschreibung der Token nicht um eine möglichst präzise grammatische Verortung der Merkmale geht, sondern darum, bei der Korpusauswertung möglichst viele Merkmale der Wörter berücksichtigen zu können.

Interessant für die Weiterverarbeitung der Analyse und die Nutzung der analysierten Korpusdaten in der Korpussuche und -auswertung ist der Aspekt, dass in der obigen Darstellung sämtliche Flexionsmerkmale aneinandergereiht werden. Dies hat zur Folge, dass wir sehr komplexe Tags erhalten können, in denen sich später bei der Korpussuche einzelne Merkmale nur mithilfe von regulären Ausdrücken (s. Kap. 3.1.2.3) erfassen lassen. In modernen Korpusarchitekturen werden die einzelnen Informationen oftmals auf gesonderte Analyseebenen geschrieben, die jeweils nur einen Typ von Information beinhalten. Dieser Logik nach kann man die in (1) gezeigte Inline-Annotation verlustfrei durch die in Tab. 2.1 gezeigte Mehrebenenannotation abbilden.

Die Sternchen in Tab. 2.1 bedeuten, dass eine Variable für die gegebene Wortart zwar gilt, aber im aktuellen Kontext nicht angegeben werden kann; die Striche bedeuten, dass die jeweilige Variable für die gegebene Wortart nicht gilt.

Bei einer solchen Datenarchitektur kann man später bei der Auswertung der Korpusdaten elementar nach Variablen-Wert-Paaren suchen.

Automatische Annotationswerkzeuge: Wie bei den bereits behandelten Kategorien der Lemmata und Wortarten gilt, dass die zu vergebene Kategorie eine grundlegende, bei Korpussuchen häufig sehr nützliche Information darstellt, die mittels gewisser Kontextinformationen automatisch relativ akkurat erkannt werden kann. Mittlerweile existieren einige Programme, die – häufig neben anderen Tagging-Ergebnissen – flexionsmorphologische Kategorien ausgeben. Dies ist u. a. der Fall bei Helmut Schmids RFTagger (http://www.cis.uni-muenchen.de/~schmid/tools/RFTagger/), der zusätzlich zu Wortartkategorien Flexionsmerkmale ausgibt. Dies geschieht auf derselben Analyseebene mithilfe von aneinandergereihten Tags (als Separator dient jeweils ein Punkt), so dass bei der Korpussuche die einzelnen Werte (z. B. »Nom« für Nominativ) nicht isoliert abgefragt, sondern z. B. mit regulären Ausdrücken (s. Kap. 3.1.2.3)

erfasst werden müssen. Der »Stuttgart Morphological Analyzer« (SMOR; https://bit.ly/2HK6tMZ) gibt morphologische Kategorien auf jeweils getrennten Analyseebenen aus, so dass einzelne Werte (wie »nominative«) auf den entsprechenden Analyseebenen (hier: »case«) isoliert gesucht werden können. Diese beiden Ressourcen sind in der Verarbeitungskette des online nutzbaren Annotationsportals WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblicht/; s. Kap. 2.3) nutzbar.

Arbeitsaufgaben

1. Reichern Sie die Datei, die Sie unter der Webadresse https://bit.ly/2HK6PDj beziehen können, mit flexionsmorphologischen Werten an (die ersten zwei Sätze sind bereits beispielhaft annotiert).
   - Sie können die Datei in die Tabelleneditoren von LibreOffice (oder OpenOffice) Calc oder Microsoft Office Excel einlesen.
   - Verwenden Sie hierbei die Richtlinien von Crysmann et al. 2005 (https://bit.ly/2Hwj9rL).
   - Speichern Sie das Ergebnis mit dem Zusatz »_getaggt_auto« im xlsx-Format.

2. Taggen Sie die Datei, die Sie unter der Webadresse https://bit.ly/2CxEr3R herunterladen können, mit flexionsmorphologischen Kategorien, indem Sie in WebLicht den RFTagger darauf anwenden.
   - Lesen Sie hierfür die Datei »Tokenisieren.txt« ein (Funktion »Upload a file: Browse«).
   - Spezifizieren Sie die Sprache (wählen Sie »German«).
   - Wählen Sie den einfachen Modus (»Easy Mode«).
   - Wählen Sie »Morphology«.
   - Klicken Sie auf »Run Tools«.
   - Exportieren Sie die Analyse mit der Funktion »Download as Excel sheet«.
   - Nutzen Sie ggf. auch die Informationen aus Kap. 2.3.
   - Speichern Sie das Ergebnis unter dem Namen »Tagging_Flexionsmorphologie_getaggt_auto.xlsx«.

3. ~~(Zusatzaufgabe:)~~ Sie können nun die beiden Analysen vergleichen, indem Sie die Spalte mit den Tagginginformationen aus der automatisch analysierten Datei neben die Spalte mit den flexionsmorphologischen Werten aus der manuell annotierten Datei kopieren (Sie werden möglicherweise an zwei Stellen alignieren – in Deckung bringen – müssen, weil die Tokenisierung des in WebLicht verwendeten Tokenisierers nicht mit der Tokenisierung in der manuell getaggten Datei übereinstimmt).
   Speichern Sie diese Vergleichsdatei unter dem Namen »Taggen_Flexionsmorphologie_getaggt_automatisch_manuell_Vergleich.xlsx«.

2.2.6.7 | Wortbildungsmorphologie

Als **Wortbildungsmorphologie** wird der Teil der ~~Wortbildung bzw. Wortbildungslehre~~ bezeichnet, der nicht die Flexion, sondern die Bildung neuer Wörter betrifft.

Morphologie
Definition

Die Analyse von Wortbildungsbestandteilen bezieht sich also auf die Teile komplexer Wörter, von denen mindestens ein Bestandteil ein ungebunden vorkommendes (lexikalisches) Morphem sein muss. Die übrigen Bestandteile können ebenso lexikalische oder aber gebundene Morpheme sein. Bei einer echten wortbildungsmorphologischen Analyse geht es ~~also~~ darum, komplexe Wörter in ihre Bestandteile zu zerlegen und den komplexen Einheiten eine Bezeichnung für die Art ihrer Zusammensetzung zu geben. Alle nicht komplexen, also nicht weiter zerlegbaren Formen (sog. Simplizia) werden als solche gekennzeichnet. Auf diese Weise können Korpusdaten auf bestimmte wortbildungsmorphologische Einheiten wie Komposita (aus lexikalischen Morphemen zusammengesetzte Wörter) oder Derivationsprodukte (mittels Affixe abgeleitete lexikalische Stämme) untersucht werden.

Tokenisierung in wortbildungsmorphologisch annotierten Korpora: Die Konsequenz einer solchen Betrachtung ist eine Verschiebung der Token-Definition hin zur linguistischen Einheit ›Morphem‹ (= kleinste bedeutungstragende Einheit in einem Wort). Zur Erinnerung: Normalerweise wird das Token als kleinste zählbare Einheit im Korpus ~~als~~ Wort und/oder Satzzeichen definiert.

im Sinne von >

Beispiele für wortbildungsmorphologische Analysen: Reichern wir den ~~bereits mehrfach verwendeten~~ Beispielsatz morphologisch an:

<

(1) *Ich Beziehungsexperte verschicke an dich morgen eine Sendung.*

(1)

So können wir, ohne die gewohnte Wort-Tokenisierung zu verändern, jedem Token entweder den Wert für ein nicht komplexes Wort (ein Simplex) oder aber für eine spezifische Komplexitätskategorie (Kompositum, Derivat, Partikelverb, Präfixverb usw.) geben:

(2) *Ich*/SIMP *Beziehungsexperte*/KOMP *verschicke*/PRÄF-V *an*/SIMP *dich*/SIMP *morgen*/SIMP *eine*/SIMP *Sendung*/DER ./SIMP

Diese Analyse sagt aus, dass sämtliche mit »SIMP« markierten Token als Simplizia interpretiert werden und dass *Beziehungsexperte* als Komposition, *verschicke* als Präfixverb und *Sendung* als Derivation aufgefasst werden. Will man die Analyse wirklich präzise durchführen und zulassen, dass auch mehrfach komplexe Wörter adäquat interpretiert werden, muss man eine morphologisch motivierte Tokenisierung der Daten vornehmen und jedem Morphem einen eigenen Wert zuordnen sowie die Wortbildungsprodukte benennen (s. Tab. 2.2).

Marginalie: "Die Auszeichnung von Morphemen erfordert eine entsprechende Tokenisierung"

Morphem	*Ich*	*Bezieh*	*ung*	*s*	*experte*	*ver*	*schicke*
Wort	*Ich*	*Beziehungsexperte*				*verschicke*	
Morphemtyp	SIMP	V-ST	N-SUFF	FUGE	N-ST	V-PRÄF	V-ST
Komplex-E 1		DER-EXPL				PRÄF-V	
Komplex-E 2		KOMP-DET					
Morphem	*an*	*dich*	*morgen*	*eine*	*Send*	*ung*	.
Wort	*an*	*dich*	*morgen*	*eine*	*Sendung*		.
Morphemtyp	SIMP	SIMP	SIMP	SIMP	V-ST	N-SUFF	SIMP
Komplex-E 1					DER-EXPL		
Komplex-E 2							

Tab. 2.2: Abbildung der wortbildungsmorphologischen Merkmale aus Beispiel (1) als Mehrebenenannotation

Die Bearbeitung in Tab. 2.2 zeigt in der obersten Zeile eine Tokenisierung nach Morphemen, die wiederum in der darunterliegenden Zeile zu Wortformen kombiniert werden (die Anordnung der Zeilen ist generell relativ flexibel bzw. kann in den meisten Datenarchitekturen und Korpusvisualisierungen relativ problemlos umgestellt werden). In der dritten Zeile findet sich die Typisierung der Morpheme (SIMP = Simplex; V-ST = Verbstamm; N-SUFF = nominales Suffix; FUGE = Fugenelement; N-ST = nominaler Stamm; V-PRÄF = verbales Präfix; V-ST = verbaler Stamm). In der vierten und letzten Zeile findet sich die Bezeichnung der Wortbildungsprodukte (DER-EXPL = explizite Derivation; PRÄF-V = Präfixverb; KOMP-DET = Determinativkompositum). Die zweite Ebene dieser Analyseebene wird nur verwendet, wenn durch mehrere Wortbildungsprodukte im selben Wort eine Verschachtelung der Strukturen vorliegt. Deshalb ist denkbar, dass im Falle noch komplexerer Wörter weitere Ebenen hinzugefügt werden müssen.

Annotationswerkzeuge: Hinter den meisten Werkzeugen, die eine als ›morphologisch‹ bezeichnete Analyse liefern, verbirgt sich eine rein flexionsmorphologische Analyse, also keine Information über den Wortbildungsstatus der Token, geschweige denn eine morphologische Zerlegung komplexer Wörter. Mit »Morphy« (http://morphy.wolfganglezius.de; Lezius 2000) hat Wolfgang Lezius einen Tagger konstruiert, der Komposita auflösen kann. Leider ist dieses Werkzeug auf aktuellen Betriebssystemen nicht ohne weiteres ausführbar. Das Werkzeug »SMOR« (https://bit.ly/2USmhkM; Schmid et al. 2004) ist dafür konstruiert, sowohl Kompositions- als auch Derivationsbestandteile kenntlich zu machen. Verbpartikeln, sofern man sie als morphologische Bestandteile ansieht, werden von den gängigen Taggern erkannt und mit einem entsprechenden Tag (STTS: »PTKVZ«) versehen. Dies gilt allerdings nur, wenn sie im Satz getrennt vom Verbstamm stehen.

Arbeitsaufgabe

Bearbeiten Sie die unter der Webadresse https://bit.ly/2TQsUY5 befindliche EXMARaLDA-Datei zur manuellen Analyse von Wortbildungsphänomenen.

- Ziel der Analyse ist es, jedem Token (Tokendefinition: Wort oder Satzzeichen) im Text einen Wert für den letzten stattgefundenen wortbildungsmorphologischen Prozess zuzuweisen. Flexion wird dabei nicht berücksichtig. Simplizia (einfache, wortbildungsmorphologisch nicht weiter zerlegbare Wortformen) werden als solche gekennzeichnet. Fremdwörter, deren morphologischer Status vor dem Hintergrund der deutschen Wortbildung intransparent ist, ebenso. Token, die keine Wörter (sondern Satzzeichen) sind, werden nicht weiter analysiert (sie erhalten das Tag »--«).
- Wenn Sie sich selber ein wortbildungsmorphologisches Tagset ausdenken (z. B. Komposition = »KOMP«, Derivation = »DER« usw.) und dieses auf die Daten anwenden, werden Sie feststellen, dass bestimmte Wortformen schwer zuzuordnen sind und dass Sie häufig neue Kategorien hinzufügen müssen.
- Unter der Webadresse https://bit.ly/2Fe1Esk finden Sie eine Datei, die Sie mit EXMARaLDA einlesen können, so dass Ihnen wortbildungsmorphologische Werte vorgegeben werden. Gehen Sie nach dem Herunterladen der Datei wie folgt vor:
  - Klicken Sie in EXMARaLDA auf »View« > »Annotation panel«.
  - Klicken Sie auf dem Panel auf »Open«.
  - Lesen Sie die XML-Datei ein.
- Nun können Sie für jede aktive Zelle einen der vorgeschlagenen Werte einfüllen (Doppelklick auf die entsprechende Kategorie im Panel). Klicken Sie unten im Panel außerdem auf die Option »Auto jump«, um von Zelle zu Zelle springen.
- Speichern Sie die fertig bearbeitete Datei unter dem Namen »Tagging_Wortbildungsmorphologie_getaggt.exb«.

2.2.6.8 | Eigennamen

Definition

Als **Eigennamen** bezeichnet man sämtliche Wörter, die einem Lebewesen, einem Ort, einer Organisation oder einem anderen Objekt eine individuelle Bezeichnung geben. Damit sind Eigennamen gegenüber den Gattungsnamen bzw. Appellativa abzugrenzen, die die Objekte der Welt in semantische Klassen Gliedern.

Eigennamen sind eine spezielle Kategorie, die im Natural Language Processing schon lange Berücksichtigung findet. Dies liegt zum einen daran, dass die Eigennamen zwar zu den Nomina gehören, allerdings eine an-

dere Grammatik als diese besitzen (sie haben in der Standardsprache bzw. in schriftlichen Registern des Deutschen keinen Artikel und werden für gewöhnlich anders attribuiert als normale Nomina). Deshalb ist es wichtig für automatische Verarbeitungsprogramme, Eigennamen zu erkennen und von den normalen Nomina zu unterscheiden. Die korrekte Erkennung von Eigennamen ist relevant für die syntaktische Analyse, für eine etwaige semantische Analyse, für die Erkennung nominaler Bezüge bei der Diskursanalyse (s. Kap. 2.2.8), für Übersetzungsprozesse und diverse andere Anwendungen. Außerdem bieten Eigennamen im Text eine sehr spezifische Informationsquelle, da sie auf ganz bestimmte, meist prominente Entitäten verweisen und eher den Kern dessen ausmachen, worüber geredet wird, als die Peripherie. Eigennamen können verknüpft werden mit verschiedenen Datenbanken: Personenlexika, geographischen Daten usw. Aus diesen Gründen erachtet man es in der automatischen Analyse natürlicher Sprache als wichtige Aufgabe, Eigennamen im verarbeiteten Text zu identifizieren und auszuzeichnen. Das entsprechende Forschungsfeld nennt sich Named Entity Recognition (NER).

Linguistisch-semantische Kategorisierung von Eigennamen: Eigennamen können in verschiedene Bedeutungstypen unterschieden werden, die sich durch den Typus der bezeichneten Entität definiert. Im Kontext der automatischen Eigennamenerkennung werden häufig Personen-, Orts- und Organisationsnamen unterschieden und durch entsprechende Tags kenntlich gemacht. Manchmal wird über Eigennamen im engeren Sinne hinausgegangen und Datumsangaben, Uhrzeiten, Telefonnummern und andere Verweise auf temporale, lokale oder personenbezogene Informationen werden in demselben Erkennungsverfahren ebenso ausgezeichnet.

Um automatisch Personennamen zu erkennen, können verschiedene Verfahren verwendet werden, z. B. können digitale Wörterbücher oder andere Wissensressourcen genutzt werden oder die Daten per Hand annotierter Korpora, in denen also Namen von menschlichen Annotatoren aufgrund ihres Weltwissens bereits ausgezeichnet wurden.

Die automatische Zuweisung von Eigennamen ist dann technisch ein relativ einfaches Prinzip, wenn die Form des Eigennamens eindeutig dieser Wortklasse entspricht. Viele Namen sind jedoch aus normalen Nomina entstanden, wodurch häufig beide Bedeutungen – die des regulären Nomens und die des Eigennamen – koexistieren. Vergleichen Sie das folgende Beispiel aus dem TIGER-Korpus (https://bit.ly/1cmgyFw):

(1) *Kohl eröffnet Konferenz* (TIGER-Korpus v2.1, s1774)

Ohne Weltwissen ist es nicht trivial zu ermitteln, dass es sich bei *Kohl* hier um ein menschliches Individuum und nicht das Gemüse handelt. Für die Annotatorinnen und Annotatoren des TIGER-Korpus, das manuell annotiert bzw. korrigiert wurde, bestand jedoch kein Problem dieser Zuweisung, und das Ergebnis der manuellen Analysen ist im vergangenen Jahrzehnt ~~vielseitig~~ in das Training automatischer Werkzeuge wie Eigennamenerkennungsprogramme eingeflossen.

Annotation von Eigennamen: Wie bei jeder Annotation benötigt man zunächst Annotationsrichtlinien, die bei Analysezielen wie dem hiesigen

ein Tagset mit einer bestimmten Anzahl an Kategorien beinhalten. Das STTS-Tagset enthält, ohne [hrl] weiter zu differenzieren, die nominale Kategorie Eigenname (NE).

Im als Baumbank aufbereiteten Zeitungkorpus TüBa-D/Z (zu syntaktisch annotierten Korpusdaten s. Kap. 2.2.7) bzw. den dazugehörigen Annotationsrichtlinien (Telljohann et al. 2017, S. 56 f., https://bit.ly/2Y8cSYh) finden sich differenzierte Eigennamenkategorien, die in der Annotationspraxis dieses Korpus der Phrasenannotation von Nominalphrasen hinzugefügt werden. Die Kategorien und dazugehörigen Tags sind die folgenden:

Person	PER
Organisation	ORG
Ort	LOC
Geopolitische Instanz	GPE
Andere	OTH

Automatische Programme, die Eigennamen ausgeben, sind einfache Wortartentagger wie der TreeTagger, sofern man sich mit einer Klasse von Eigennamen zufrieden gibt. Auf dem bereits erwähnten Internetportal zur automatischen Verarbeitung von Korpusdaten WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblicht/) können Sie verschiedene Eigennamentagger anwenden, von denen ein speziell für das Deutsche entwickelte Eigennamenerkenner (https://bit.ly/2Fib06G, Faruqui/Pado 2010) der mit der differenziertesten Ausgabe ist. Ein weiteres Programm findet man unter https://github.com/tudarmstadt-lt/GermaNER (der an der TU Darmstadt entwickelte Eigennamenerkenner GermaNER, Benikova et al. 2015).

Arbeitsaufgabe

Überprüfen Sie die Eigennamenerkennung des TreeTaggers ([hrl] TagAnt-Version), indem Sie den [Text unter] der Webadresse https://bit.ly/2HAl1iT mit TagAnt verarbeiten (s. die Hinweise zur Nutzung von TagAnt in Kap. 2.2.6.4).

- Konvertieren Sie die von TagAnt ausgegebene Datei »Eigennamenerkennung_tagged.txt« in das Format ANSI (nutzen Sie hierfür einen fortgeschrittenen Texteditor wie Notepad + +, https://notepad-plus-plus.org/).
- Importieren Sie die Datei »Eigennamenerkennung_tagged.txt« in EXMARaLDA (über die Option »File« > »Import...« und die »Dateityp:«-Einstellung »Tree Tagger Output (*.txt)«.
- Fügen Sie eine Annotationsspur (Tier) »pos_korrigiert« hinzu (Option »Insert tier«) und übernehmen Sie die Werte der Spur »pos«.
- Korrigieren Sie die Analyse sämtlicher Eigennamen und derjenigen Token, die fälschlicherweise als Eigennamen erkannt wurden (das STTS-Tag für Eigennamen ist »NE«).
- Speichern Sie die Datei unter dem Namen »Eigennamenerkennung_TreeTagger_korrigiert.exb«.

2.2.7 | Syntaktische Kategorien: Chunking und Parsing

Definition

Chunking bezeichnet, unabhängig von der inhaltlichen Analyse, das Zusammenfügen von fortlaufenden Einheiten (meistens Token) zu größeren Bausteinen, sogenannten Chunks oder auch Spannen. Der Begriff hat sich für das Zusammenfassen von Token zu syntaktischen Phraseneinheiten wie PPn, NPn usw. etabliert.
Parsing bedeutet die vollständige Analyse, so dass jedem Wort im Satz bzw. jedem Token in der zu analysierenden Struktur ein individueller Status bezüglich seiner (syntaktischen) Eigenschaften in der Struktur zugewiesen wird. Auch das Parsing ist nicht notwendigerweise an syntaktische Analysen geknüpft.

Token-Annotationen vs. komplexere Annotationen: Nachdem die bisherigen Annotationsarten vor allem tokenbasierte Annotationen waren (sofern Wörter und Satzzeichen als Token definiert werden), betreffen die syntaktischen und textbezogenen Annotationsarten nicht nur das einzelne Token, sondern auch Abfolgen von Token, die nicht einmal kontinuierlich sein müssen. In der linguistischen Tradition gibt es verschiedenste Ansätze, die syntaktische Interpretation von Sätzen zu bewerkstelligen. In der Korpuslinguistik haben sich solche Analysen durchgesetzt, die sich sehr an dem im Satz vorhandenen sprachlichen Material orientieren, also oberflächenbasiert sind, und die eher traditionelle, theorieübergreifende funktionale Kategorien wie »Subjekt« und »Akkusativobjekt« beinhalten.

Chunking – ein Beispiel: Eine Möglichkeit der Annotation des Beispielsatzes *Ich schreibe dir morgen einen ausführlichen Brief.* mit verschiedenen Chunks zeigt Abb. 2.6.

Die Abb. 2.6 zeigt verschiedene Chunks: Auf der Ebene »cat_S« wird der gesamten Struktur eine Satzspanne zugewiesen; auf den Ebenen »cat_XP« werden Phrasen gekennzeichnet (NP = Nominalphrase, AdvP = Adverbialphrase, AP = Adjektivphrase); auf der Ebene »cat_topo« werden topologische Felder und Satzklammern markiert (VF = Vorfeld, LK = linke Satzklammer, MF = Mittelfeld).

Beim Chunking entstehen also Spannen, die eine gewisse Anzahl von Elementen abdecken. Somit kann man den Begriff ›Spanne‹ synonym zu dem Begriff ›Chunk‹ verwenden. Die Ebenen, vor allem die mit »cat_XP«

Abb. 2.6: Exemplarisches Chunking auf verschiedenen Beschreibungsebenen

[tok]	Ich	schreibe	dir	morgen	einen	ausführlichen	Brief.
[cat_S]	Satz						
[cat_XP]	NP		NP	AdvP	NP		
[cat_XP]						AP	
[cat_topo]	VF	LK	MF				

bezeichneten, zeigen die Grenzen des Chunkings: Sobald Phänomene komplex werden, indem sich Strukturen überlagern bzw. hierarchisch aufgebaut sind, genügt eine einzelne Beschreibungsebene nicht.

Parsing – ein Beispiel: Eine Möglichkeit der Annotation des Beispielsatzes *Ich schreibe dir morgen einen ausführlichen Brief.* mit einem Parse zeigt Abb. 2.7.

Die Abb. 2.7 zeigt die Struktur des gesamten Satzes (S) mit direkten Kanten zu den Einwortkonstituenten sowie der komplexen Nominalphrase (NP) *einen ausführlichen Brief*. Die Kantenbezeichnungen stehen für: SB = Subjekt, HD = Head (Kopf der Struktur), DA = Dativkonstituente, MO = Modifikator, OA = Akkusativobjekt.

Da man Sätze auf ganz unterschiedliche Weise vollständig syntaktisch analysieren kann, ist die Analyse in Abb. 2.7 nur ein Beispiel für Parsing unter vielen. Auch schließen Chunking und Parsing einander nicht aus: Es ist möglich, einen Satz mithilfe von Chunks vollständig (wortweise) zu analysieren, sofern man einen einfachen Satz vorliegen hat. In diesem Fall entspricht ein Chunking dann auch der Definition von Parsing.

Für viele linguistische Analysen sind syntaktische Kategorien eine wichtige Grundlage, und ein großer theoretischer Forschungsbereich basiert auf syntaktischen Analysen. So ist der Wunsch nach der Anreicherung von Korpusdaten mit syntaktischen Kategorien absolut essenziell. Auf diese Weise können Wörter und Wortgruppen in bestimmten syntaktischen Positionen und Funktionen gefunden werden. Zusammengefasst ist es normalerweise das Ziel einer syntaktischen Analyse, die folgenden grammatischen Aspekte aufzuzeigen:

- die Zusammengehörigkeit von Wörtern zu Konstituenten im Satz,
- die hierarchische Darstellung von kleineren und größeren Konstituenten im Satz,
- die Funktion der einzelnen Elemente für ihre Mutterkonstituenten sowie
- die Funktion der tiefer im Satz liegenden Konstituenten hinsichtlich der nächsthöheren Konstituente.

Dies muss natürlich nicht vollständig, also unter Berücksichtigung aller im Satz vorhandenen Funktionseinheiten, erfolgen. Außerdem können

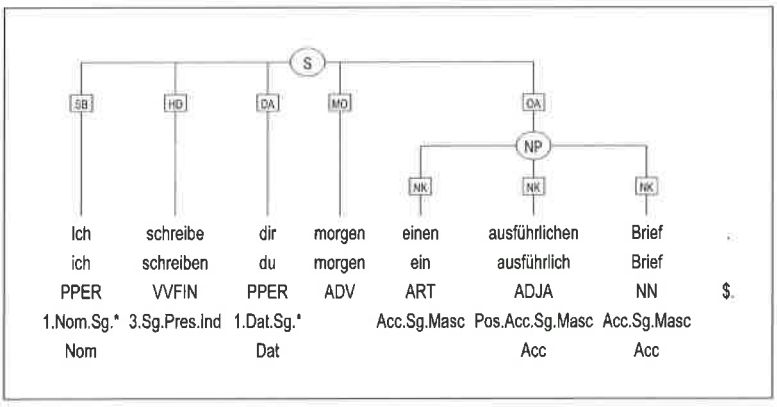


Abb. 2.7: Beispiel für einen Phrasenstrukturparse

~~syntaktische Analysen mit verschiedensten theoretischen Annahmen verknüpft sein~~ Die nachfolgenden Abschnitte beschäftigen sich genauer mit verschiedenen syntaktischen Analyseverfahren, die entweder dem Chunking oder dem Parsing zuzuordnen sind.

2.2.7.1 | Satzspannen

Definition

> Eine **Spanne** ist eine fortlaufende Abfolge von Token im Korpus, der eine linguistische Kategorie zugeordnet wird. Die Spanne ist somit eine bestimmte Annotationsart, gleichbedeutend mit ›Chunk‹, der Begriff ist aber konzeptionell weniger auf die Phrasenanalyse beschränkt. **Satzspannen** sind ein bestimmter Typ der Spannenannotation.

Wie bereits in Abb. 2.6 gezeigt, kann mittels Chunking die Analyse von Sätzen erfolgen. Wie beim Wortartentagging (s. Kap. 2.2.6.2) werden hier meistens automatische Verfahren angewendet. Dies ist technisch nicht aufwändig, weil die Information über Satzgrenzen indirekt vom Tagger ausgegeben wird, da das STTS-Tagset und andere Wortartentagsets Werte für satzbeendende Interpunktion beinhalten. Die explizite Markierung der Spanne zwischen einem satzbeendenden Token und dem nächsten, erfordert also nach durchgeführtem Tagging keine gesonderte Analyse mehr. Dies gilt zumindest für die Analyse von Gesamtsätzen, die eine kontinuierliche Segmentierung der im Korpus verarbeiteten Textdaten auf einer Analyseebene bedeutet.

Programme zur Satzsegmentierung: In dem Korpusverarbeitungsservice WebLicht (http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/, s. Kap. 2.3) werden verschiedene Satzsegmentierungsprogramme angeboten, die einer solchen Definition von Gesamtsätzen folgen:

- OpenNLP Sentence Detection (https://bit.ly/2UMUB0z)
- Sentence Splitter (Jurish/Würzner 2013; implementiert u. a. im WASTE-Tokenizer, http://kaskade.dwds.de/waste/)
- Tokenizer (als Teil der Corpus Workbench, CWB, https://bit.ly/2FrintI).

Diese Werkzeuge können in WebLicht angewendet werden. Unter http://kaskade.dwds.de/waste/demo.perl wird eine Online-Demoversion des WASTE-Tokenizers angeboten, in der sich kürzere Texte als Textdateien (.txt) oder aus der Zwischenablage einlesen und verarbeiten lassen.

Andreas Nolda hat außerdem Add-Ins für EXMARaLDA geschrieben, mit denen man untokenisierte Textdaten innerhalb von EXMARaLDA tokenisieren, taggen und nach Sätzen segmentieren kann. Die Programmdateien und Installationshinweise sind unter der Webadresse https://bitbucket.org/nolda/exmaralda-dulko/ zu beziehen.

Komplexere Auszeichnung von Satztypen: Natürlich ist die Annotation von Sätzen hochgradig abhängig von der Satzdefinition, die je nach Analysehintergrund ganz unterschiedlich ausfallen kann. In der Gramma-

tikanalyse des Deutschen ist vor allem die Unterscheidung zwischen Haupt- und Nebensätzen, abhängigen Nebensätzen (Ergänzungssätze) und unabhängigen Nebensätzen (Adverbialsätze und Attributsätze) sowie diversen semantischen Kategorien von selbständigen und unselbständigen Sätzen etabliert. Normalerweise werden solche Unterscheidungen jedoch nur im Rahmen der vollständigen syntaktischen Analyse, vor allem bei Phrasenstrukturbaumbanken (s. Kap. 2.2.7.6) ausgewiesen.

Ein Korpusbeispiel mit differenzierter Ausweisung von Satztypen: Eine erwähnenswerte Ausnahme ist das Korpus »Frühneuzeitliche Fürstinnenkorrespondenz im mitteldeutschen Raum« (https://bit.ly/2U5nzeM), in welchem diverse Satztypen voneinander unterschieden werden (Korpusdokumentation: https://bit.ly/2CvVl2N). Konkret sind die Satztypen auf einer Annotationsebene »clause-st« (für »clause status«, »Satzstatus«) annotiert, auf der jeder Teilsatz einen eigenen Wert zugewiesen bekommt. Die im Korpus häufigsten Werte sind Deklarativsätze (»decl«), Kausalsätze (»caus«), Konditionalsätze (»cond«) und Desideративsätze (Wunschsätze, »desi«). Das Korpus ist im ANNIS-Suchinterface (https://hu.berlin/annis) verfügbar; Informationen zur Nutzung erhalten Sie in Kapitel 3.1.2 f.

Arbeitsaufgabe

Unter der Webadresse https://bit.ly/2Wee8ai finden Sie eine Version des bereits mehrfach verarbeiteten Beispieltextes mit annotierten Satzspannen (Ganzsätze). (Die Verarbeitung erfolgte mittels der o. g. Programmversion von Andreas Nolda.)

- Fügen Sie auf der noch unverarbeiteten Annotationsebene für Nebensätze allen untergeordneten Sätzen Werte für ihren syntaktischen Status zu. Hierzu müssen die entsprechenden Sätze zunächst durch Spannen gekennzeichnet und die Spannen anschließend mit Kürzeln (Tags) für den syntaktischen Status versehen (»gelabelt«) werden. Die möglichen Kategorien sind: Ergänzungssatz – Objektsatz (»SOBJ«); Ergänzungssatz – Subjektsatz (»SSUBJ«); Adverbialsatz (»SADV«); Attributsatz (»SATTR«).

  Unter der Webadresse https://bit.ly/2OhE4iv können Sie eine XML-Datei beziehen, die Sie in das Annotationspanel von EXMARaLDA einlesen können. Beachten Sie ggf. die Hinweise in der Arbeitsaufgabe in Kap. 2.2.6.7 zur Konfiguration des Annotationspanels.
- Speichern Sie die fertig analysierte Datei unter dem Dateinamen »Satzspannen_Nebensatz_annotiert.exb«.

2.2.7.2 | Topologische Felder

Die Analyse von topologischen Feldern (den Stellungsfeldern »Vorfeld«, »Mittelfeld« und »Nachfeld«) und der die Felder begrenzenden Satzklammer (aufgeteilt in linke und rechte Satzklammer) gewinnt innerhalb der Beschreibung der Syntax des Deutschen zunehmend an Bedeutung. Dies liegt daran, dass das Deutsche bezogen auf die Stellung der Prädikatsteile relativ rigide ist. Aus diesem Grund ist die Beschreibung dieser Stellungssystematik nicht nur für sich sinnvoll, sondern ist auch ein wesentlicher Bestandteil von allgemeinen syntaktischen Grammatikmodellen für das Deutsche wie der X-Bar-Syntax. Auch lassen sich am Erwerb der grundlegenden Stellungsmuster wesentliche Meilensteine des Grammatikerwerbs im Deutschen festmachen.

Die Beschreibung der Stellungsfelder im Deutschen geht auf Erich Drach zurück, dessen Ausführungen in Drach (1937) erstmals erschienen sind. Heute gehört die einheitliche Analyse von Sätzen nach dem in Tab. 2.3 gegebenen Muster zu einer der Standardbeschreibungen der deutschen Grammatik.

Vorfeld (VF)	linke Satzklammer (LSK)	Mittelfeld (MF)	rechte Satzklammer (RSK)	Nachfeld (NF)
Ich	*schreibe*	*dir morgen einen ausführlichen Brief.*		
Ich	*habe*	*dir gestern einen ausführlichen Brief*	*geschrieben.*	
Gestern	*habe*	*dir einen ausführlichen Brief*	*geschrieben,*	*weil ich dir so alles besser erklären kann.*
	weil	*ich dir so alles besser*	*erklären kann.*	
	Kannst	*du es mir bitte*	*erklären?*	
	Erkläre	*es mir bitte,*		*nachdem wir gegessen haben!*

Tab. 2.3: Zusammenfassende Sprachbeispiele für die Analyse topologischer Felder

Höhle (1986) bietet eine kritische Zusammenfassung des Analyseansatzes. Beachten Sie auch die Darstellungen in Schäfer (2016a, S. 411 f.).

Die Analyse topologischer Felder in der Korpuslinguistik: In der Korpuslinguistik ist das Konzept der Analysen von topologischen Feldern voll und ganz etabliert, indem sowohl umfassende Richtlinien zur Erstellung entsprechender Annotationen als auch automatische Verfahren und Korpora mit in der Auswertung nutzbaren Annotationen existieren.

Annotationsrichtlinien: Die einflussreichsten Richtlinien zur Beschreibung toplogischer Felder und Klammerstrukturen finden sich in dem »Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)« (Telljohann et al. 2017, https://bit.ly/2TQpGDX). Nach diesen Guidelines ist das gesamte TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL) entstanden (s. auch Kap. 2.2.7.6), welches Informationen über topologische Felder und Satzklammerstrukturen mit Informationen zu syntaktischen Phrasenstrukturen in einem Baum abbildet.

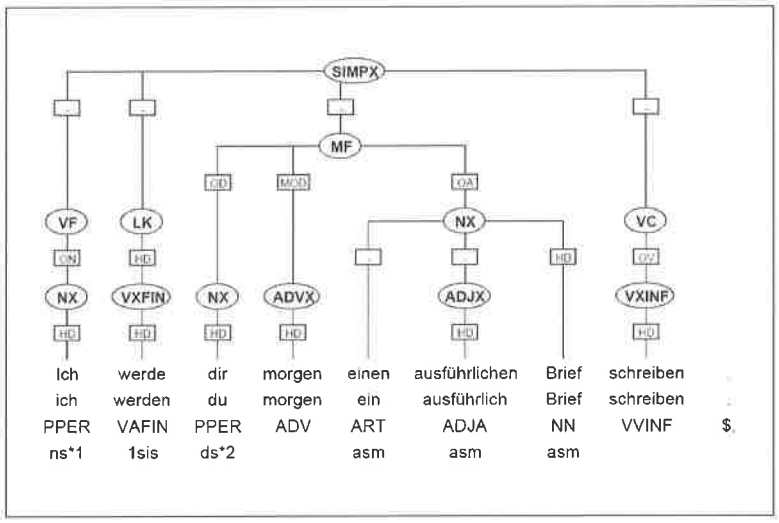


Abb. 2.8: Der Beispielsatz *Ich werde dir morgen einen ausführlichen Brief schreiben* im TüBa-D/Z-Format mit topologischen Informationen (die relevanten ~~topologischen~~ Kategorien von links nach rechts): VF = Vorfeld, LK = linke Satzklammer, MF = Mittelfeld, VC = Verbalkomplex bzw. rechte Satzklammer

Doolittle (2008) formuliert ebenso Richtlinien für die Felderannotation und evaluiert diese kritisch (~~https://edoc.hu-berlin.de/handle/18452/14786~~).

Annotationswerkzeuge: Der Berkeley-Parser (https://github.com/slavpetrov/berkeleyparser) gibt mit der Parameterdatei bzw. der Grammatik für das Deutsche (»ger_sm5.gr«) syntaktische Bäume nach dem TüBa-D/Z-Schema aus. Der »Topoparser« (Cheung/Penn 2009, https://bit.ly/2Fu8dsp) ist eine Adaptation der Grammatik des Berkeley-Parsers, die aus der gesamten TüBa-D/Z-Grammatik lediglich die topologischen Informationen und einige Phrasenmerkmale berücksichtigt und auf syntaktische Funktionen verzichtet. Beide Ressourcen sind im NLP-Verarbeitungsportal WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblicht/) verfügbar.

Korpora: Wie bereits erwähnt, enthalten die mit dem TüBa-D/Z-Annotationsschema annotierten Korpora Informationen über topologische Felder. Diese Korpora sind das TüBa-D/Z-Korpus selbst (https://bit.ly/2WehXMO), aber auch das spontansprachliche Pendant TüBa-D/S (https://bit.ly/2uiLkSa) und weitere Derivate (eine Zusammenfassung bietet die Webseite https://bit.ly/2TPLLm1).

Weil sich im Bereich der Lernersprache des Deutschen als Fremdsprache Grammatikalitätsprobleme im Bereich der Wortstellung mittels topologischer Analysen ideal beschreiben lassen, enthalten einige Lernerkorpora Annotationen topologischer Felder. Beispiele sind das Kobalt-DaF-Korpus (https://hu.berlin/kobalt-daf) sowie das Falko-Summary-Korpus (zusammenfassende Webseite zu den Falko-Korpora: https://hu.berlin/falko-design; Beschreibung der Annotationen im Summary-Subkorpus: https://hu.berlin/falko-summary-topo-guide).

Arbeitsaufgabe

- Laden Sie die unter der Webadresse https://bit.ly/2TWCOXB verfügbare Datei herunter und öffnen Sie diese in EXMARaLDA (Doppelklick bzw. »File« > »Open...« im Partitureditor).
- Annotieren Sie die Datei, indem Sie den Sätzen auf der Annotationsebene »Topologische Felder« die Kategorien »VF« für »Vorfeld«, »LSK« für »linke Satzklammer«, »MF« für »Mittelfeld«, »RSK« für »rechte Satzklammer« und »NF« für »Nachfeld« zuweisen.
  - Orientieren Sie sich bei der Vergabe der Kategorien an der Übersicht in Tab. 2.3 und ziehen Sie Sie ggf. die in diesem Kapitel referenzierte Literatur zu Rate.
  - Unter der Webadresse https://bit.ly/2JvfzQo können Sie eine XML-Datei beziehen, die Sie in das Annotationspanel von EXMARaLDA einlesen können. Beachten Sie ggf. die Hinweise in der Arbeitsaufgabe in Kap. 2.2.6.7 zur Konfiguration des Annotationspanels.
- Speichern Sie die bearbeitete Datei unter dem Namen »Topologische_Felder_annotiert.exb«.

2.2.7.3 | Phrasenanalyse: Chunking von Maximalphrasen (*shallow parsing*)

Definition

> ***Shallow parsing*** bezeichnet einen selektiven Parsingprozess, bei dem lediglich die Phrasenkategorien der Satzkonstituenten analysiert werden. Diese Konstituenten werden auch als **Maximalphrasen** bezeichnet.

Will man auf ein aufwändiges Parsen verzichten und möchte dennoch Informationen über syntaktische Phrasen im Korpus speichern, so bietet sich das Chunking bzw. die Annotation mit Phrasenspannen an. Da sich auf einer bestimmten Betrachtungsebene (z. B. einer Annotationsebene für Phrasen) aber nicht sämtliche Phrasen im Satz kennzeichnen lassen, weil Phrasen ineinander verschachtelt sein können, muss ein Phrasenchunking immer selektiv erfolgen. Das bedeutet, nur bestimmte Phrasen werden markiert, andere werden weggelassen.

Fallbeispiel: Stellen Sie sich vor, Sie führen ein Phrasenchunking in einem Satz mit der Präpositionalphrase (PP) *auf das kleine Kind* durch. Wollen Sie die PP als einen zusammenhängenden Baustein kennzeichnen, können Sie die enthaltene Nominalphrase (NP) schwer gesondert kennzeichnen. Dasselbe gilt für die in der NP enthaltene Adjektivphrase. Sie können natürlich Richtlinien formulieren, gemäß deren Sie bei PPn nur den Kopf markieren o. Ä. Nun sind Sie jedoch nicht mehr in der Lage, in dem gegebenen Beispiel die Wörter *kleine* und *Kind* eindeutig als zur

PP zugehörig zu markieren (eine PP kann nach der ersten nominalen Einheit, z. B. einem Pronomen *das*, ihre rechte Grenze haben). Dies illustriert, dass flache Kennzeichnungsverfahren nur bedingt dazu geeignet sind, Syntax umfassend abzubilden.

Chunking von Maximalphrasen (*shallow parsing*): Eine Lösung für das dargestellte Problem ist die Entscheidung, bezogen auf einen Phrasenkopf mit Satzgliedstatus jeweils die größtmögliche Phrase auszuzeichnen. Unter der Voraussetzung, dass das oben genannte Beispiel *auf das kleine Kind* ein Satzglied darstellt (also ein Präpositionalobjekt oder ein Adverbial ist) bedeutet dies, dass die gesamte Struktur als PP gekennzeichnet wird und auf die Auszeichnung der Nominalphrase (*das kleine Kind*) und der Attributphrase (Adjektivphrase, *kleine*) verzichtet wird. Verbalphrasen und höher liegende Phrasen im Satz werden bei diesem Ansatz nicht im Gesamten berücksichtigt, weil dies wiederum die Tilgung der nominalen und präpositionalen Phrasen aus der Analyse bedeuten würde. Es findet jedoch eine Markierung der verbalen Phrasenköpfe (*Peter* $_{VP}$[*geht*] *nach Hause*) sowie die von Verbalkomplexen (..., *dass Peter das* $_{VC}$[*machen will*]) statt. Generell hat dieser Ansatz zur Folge, dass satzgliedwertige Nominalphrasen (NP), Präpositionalphrasen (PP), Adjektivphrasen (AP) und Adverbialphrasen (AdvP) sowie die Prädikatsteile gekennzeichnet werden und die Analyse sämtliche Wörter im Satz abdeckt. ›Satzgliedwertig‹ bedeutet ›unmittelbare Satzkonstituente‹ (vgl. Schäfer 2016a, S. 353). ~~Dieses Verfahren ist auch als~~ ›shallow parsing‹ (flaches Parsen) ~~bekannt und~~ wird in dem Aufsatz Abney (1991) beschrieben.

Arbeitsaufgabe

- Laden Sie die unter der Webadresse https://bit.ly/2TVPrlG verfügbare Datei herunter und öffnen Sie diese in EXMARaLDA (Doppelklick bzw. »File« > »Open...« im Partitureditor).
- Annotieren Sie die Datei, indem Sie zunächst Spannen für die satzgliedwertigen Phrasen und verbalen Cluster ziehen und anschließend Phrasenkategorien hinzufügen. Halten Sie sich bei der Phrasenanalyse an die o. g. Hinweise zur Analyse von Maximalphrasen.
- Nutzen Sie für das Auszeichnen (Labeln) die unter der Webadresse https://bit.ly/2FrAfVm beziehbare XML-Datei für das EXMARaLDA-Annotationspanel: Aktivieren Sie die Funktion »Auto jump« und nutzen Sie im Fall von nicht auszufüllenden Zellen (z. B. Satzzeichen) die Option »WEITERKLICKEN«.
- Speichern Sie die bearbeitete Datei unter dem Namen »Phrasenchunking_annotiert.exb«.

2.2.7.4 | Syntaktische Annotationen im Baumformat

Definition

Bäume bilden hierarchische Strukturen ab, meistens komplexe Syntagmen.
Knoten sind die Verzweigungsstellen in Bäumen. Ein Knoten kann z. B. eine bestimmte syntaktische Phrase repräsentieren. Er erhält dann die entsprechende Phrasenkategorie.
Kanten sind die Verbindungen zwischen Knoten. In einem Knoten laufen typischerweise mehrere Kanten zusammen. Es gibt aber auch sog. unäre Knoten, in denen nur eine Kante mündet, oder die sog. Blattknoten, die die unteren Enden der Baumstruktur darstellen.

Aus den im vorigen Kapitel genannten Gründen der beschränkten Abbildungsmöglichkeiten durch Chunks bzw. Spannen haben sich in der Beschreibung syntaktischer Strukturen Baumformate durchgesetzt. Es existieren ganze Korpora mit syntaktischen Bäumen, die man systematisch durchsuchen kann. Diese Korpora werden **Baumbanken** bzw. engl. **Treebanks** genannt. Das Ziel einer solchen Baumbank ist es, jedem Satz (auch komplexen Sätzen mit mehreren Teilsätzen) genau einen syntaktischen Baum zuzuweisen. Ein Baum definiert sich durch seine Wurzel (engl. *Koot*), die über beliebig viele Verzweigungen (Knoten; engl. *node*) mit den sogenannten terminalen Knoten, den Token bzw. Wortformen verbunden sind (s. Abb. 2.9).

Der in Abb. 2.9 dargestellte Baum dient der Veranschaulichung der Terminologie: Grundlegend besteht der Baum aus Knoten und Kanten, wobei die mit »PX«, »PY« und [von links nach rechts] mit »xy« bis »yt« bezeichneten Stellen im Baum Knoten sind und die mit »EA« bis »EH« bezeichneten Elemente Kanten sind. Alle Knoten haben gemeinsam, dass sie Verzweigungen innerhalb der Baumstruktur sind, wobei die untersten Elemente als die Enden der Verzweigungsstruktur (sog. terminale Knoten oder Blattknoten) zu interpretieren sind, der oberste Knoten (PX) als der Beginn der Verzweigungsstruktur (der. sog. Wurzelknoten, engl. *root node*) zu interpretieren ist und mit dem zwischen der Wurzel und den

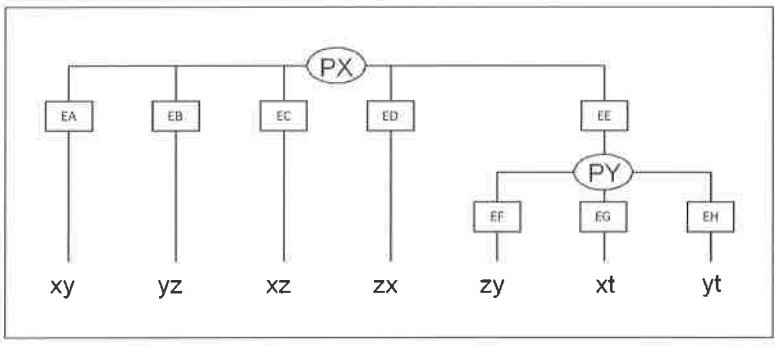


Abb. 2.9: Abstrakter Strukturbaum mit Knoten und Kanten

terminalen Knoten liegenden Knoten (PY) zusammen auch als nicht terminaler Knoten (engl. *non-terminal node*) bezeichnet wird. Die Knoten im Baum symbolisieren Entitäten (Schnittstellen) in der Struktur, die Kanten symbolisieren die Beziehung zwischen diesen Stellen. Die Verbindungskanten sowie die Knoten können Bezeichnungen tragen oder unbenannt bleiben – hierdurch wird die grundlegende Aussage der Baumstruktur nicht beeinflusst, sie wird nur unterspezifiziert. Schaut man auf die durch die Kanten ausgedrückte Relation zwischen den Knoten, so ergeben sich die folgenden Beziehungstypen:

- Höher im Baum liegende Knoten sind **Mütter** bzw. **Mutterknoten**.
- Tiefer im Baum liegende Knoten sind **Töchter** bzw. **Tochterknoten**.
- Die Beziehung zwischen Müttern und Töchtern kann unmittelbar sein (man spricht von **direkter Dominanz**) oder mittelbar (sie kann sich über beliebig viele Generationen erstrecken; man nennt dies **indirekte Dominanz**).
- An derselben Einbettungstiefe liegende Knoten sind **Geschwister** oder **Schwesterknoten**.

Somit ist »PX« in Abb. 2.9 die Wurzel und die Mutter von acht Töchtern – fünf unmittelbaren und drei mittelbaren. »PY« besitzt einen Mutterknoten, drei Tochterknoten und keinen Schwesterknoten. Die sieben terminalen Knoten »xy« bis »yt« sind Geschwister. Sie besitzen zwei verschiedene unmittelbare Mütter.

Dependenzstrukturbäume vs. Konstituenten- bzw. Phrasenstrukturbäume: Die Knoten bzw. Schnittstellen im Baum von Abb. 2.9 werden bei syntaktischen Beschreibungen als Phrasen bzw. Konstituenten innerhalb von Sätzen interpretiert, der Wurzelknoten symbolisiert den gesamten Satz. Ähnliche Konstituentenstrukturbäume kennen Sie wahrscheinlich durch generative Beschreibungstraditionen wie der X-Bar-Syntax: Das gegebene Phrasenstrukturformat kennt eine bestimmte Anzahl von Phrasenkategorien, die sich z. B. aus lexikalischen Kategorien ableiten (eine Präposition bildet eine Präpositionalphrase usw.) und bestimmte Regeln zum Aufbau der Phrasenstrukturen. Man kann Sätze (und andere sprachliche Strukturen aber auch beschreiben, indem man nach der Abhängigkeiten aller terminalen Elemente (z. B. aller Wörter im Satz) zueinander fragt und höher liegende Strukturen wie Phrasen bzw. komplexere Satzkonstituenten ausblendet. Was übrig bleibt, sind sogenannte Dependenzstrukturen nur mit terminalen Knoten und direkten Beziehungen (Kanten) unter ihnen (s. Abb. 2.10).

Auch in der in Abb. 2.10 gezeigten Struktur gibt es Knoten und Kanten, wobei auf die Kantenbezeichnung verzichtet wurde und jeder Knoten maximal eine Mutter besitzt. Die Pfeilrichtung markiert die Beziehung von der Mutter hin zur Tochter. Der Knoten ohne Mutter ist der Wurzelknoten. Alle von diesem Knoten wegverzweigenden Knoten sind dessen unmittelbare Töchter, alle weiterverzweigenden Knoten sind mittelbare Töchter. Auch hier ist die Anzahl der Generationen theoretisch unbegrenzt.

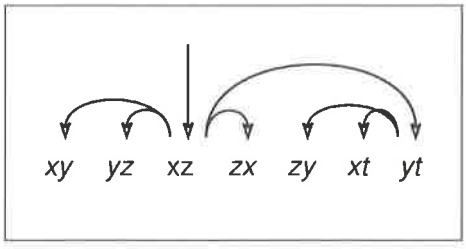


Abb. 2.10: Abstrakter Dependenzstrukturbaum mit terminalen Knoten und Kanten

Wie diese zwei grundlegenden Abbildungsformate linguistisch eingesetzt werden, wird nachfolgend beschrieben.

2.2.7.5 | Syntaktische Bäume: Dependenzstrukturen

Eine syntaktische Interpretation eines gegebenen Satzes hinsichtlich seiner Dependenzstruktur ist es, jeder im Satz enthaltenen Wortform eine Position in einem grammatischen Abhängigkeitsgefüge zuzuweisen. Vereinfacht ausgedrückt, handelt es sich hierbei um die Erweiterung von Lucien Tesnières Valenzkonzept (Tesnière 1980), bei dem der Satz ausgehend vom lexikalischen Verb als strukturelles Zentrum des Satzes beschrieben wird: Die Konstituenten um das Verb (die sog. Aktanten) werden als vom Verb gefordert oder als vom Verb unabhängig interpretiert.

Der Satz *Ich schreibe dir morgen einen ausführlichen Brief* würde nach diesem Ansatz das strukturelle Zentrum *schreibe* enthalten, welches die drei Aktanten *Ich*, *dir* und *einen ausführlichen Brief* vorsieht (fordert) und das zusätzlich von dem unabhängigen Aktanten *morgen* modifiziert wird. Für eine komplette Interpretation des Satzes fehlt dann noch die Analyse der internen Struktur von *einen ausführlichen Brief*. Auch wenn verschiedene syntaktische Theorien hierauf verschiedene Antworten geben, ist es dennoch relativ intuitiv nachvollziehbar, wenn der lexikalische Kern *Brief* als Kopf (Hauptbestandteil) interpretiert wird, der unabhängig voneinander die zwei Elemente *einen* und *ausführlichen* dominiert, wobei es mit *einen* in einem Determinierungsverhältnis und mit *ausführlichen* in einem Attribuierungsverhältnis steht.

Diese verbal ausformulierte und in Abb. 2.11 visualisierte Gesamtbeschreibung deckt sich mit der für Korpusdaten entwickelten Dependenzgrammatik von Kilian Foth (2006). Diese Grammatik ist als Inventar an Regeln zur Hierarchisierung der Elemente im Satz zu verstehen (vgl. in Abb. 2.11: Der Artikel wird als dem Nomen untergeordnet interpretiert), welches ebenso ein Inventar an Beziehungstypen umfasst (vgl. in Abb. 2.11: Der Artikel steht mit dem Nomen in einer Determinierungsbeziehung, die mit »DET« bezeichnet wird).

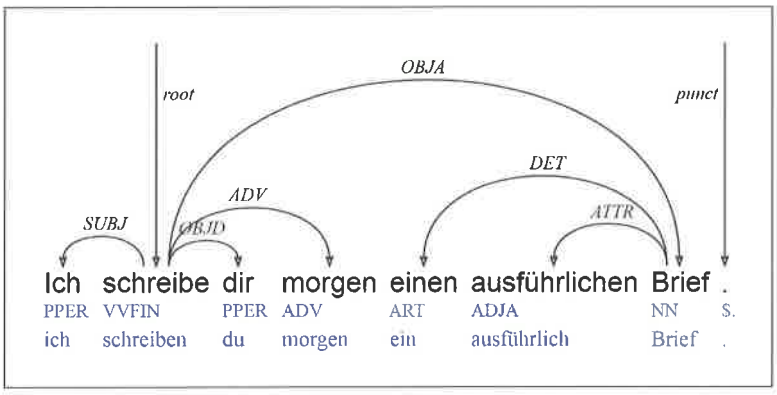


Abb. 2.11: Dependenzstrukturbaum zu dem Satz *Ich schreibe dir morgen einen Brief* gemäß Foth 2006

Abb. 2.11 zeigt zusammengefasst einen tokenisierten Satz mit Wortartentags gemäß dem STTS-Tagset und einer syntaktischen Interpretation gemäß dem Syntaxschema von Foth (2006). Hierbei werden die Satzkonstituenten dem lexikalischen Verb untergeordnet und mit den Funktionsbezeichnungen (Kantenbezeichnungen) »SUBJ« (für »Subjekt«: das Element *Ich*), »OBJD« (für »Dativobjekt«: das Element *dir*), »ADV« (für »Adverbial«: das Element *morgen*) und »OBJA« (für »Akkusativobjekt«: das Element *Brief* mit seinen Dependenten *einen* und *ausführlichen*) versehen. Die Beziehung zwischen *Brief* und *einen* ist mit »DET« für »Determinierung« und die Beziehung zwischen *Brief* und *ausführlichen* mit »ATTR« für »Attribut« ausgezeichnet. In einem auf diese Weise annotierten Text können z. B. Elemente mit demselben funktionalen Status (z. B. alle Akkusativobjekte) oder demselben dependenziellen Status (z. B. alle vom Nomen dominierte Adjektive) systematisch gefunden werden.

Manuelle und automatische Annotation von Dependenzstrukturen: Um solche Annotationen zu erstellen, kann man entweder auf automatische Programme, sogenannte Parser, zurückgreifen oder per Hand annotieren. Der effizienteste Weg zur sauberen Analyse ist bei größeren, aber noch manuell handhabbaren Datenmengen sogar eine Kombination aus beiden Ansätzen: automatisches Parsing und anschließende manuelle Korrektur.

Programme für das manuelle Bearbeiten von Korpusdaten mit Dependenzstrukturen: Im Allgemeinen kann man dort Rohdaten oder bereits automatisch vorverarbeitete Textdaten einlesen, visualisieren und Kanten zwischen den Token erzeugen und bezeichnen oder falsch analysierte Kanten umhängen oder umbenennen. Hierfür eignen sich die Programme Arborator (https://arborator.ilpga.fr/; u. a. online nutzbar; Abb. 2.11 wurde mit diesem Werkzeug erstellt) oder das auf Brat (http://brat.nlplab.org/) basierende Programm WebAnno (https://webanno.github.io/webanno/).

Programme für das automatische Parsing von Textdaten mit Dependenzen: Die Desktop-Programme (zur Verwendung auf dem eigenen Computer) sind ausschließlich kommandozeilenbasiert, besitzen also keine grafische Nutzeroberfläche und sind nur mit etwas Erfahrung in der Ausführung von Kommandozeilenprogrammen bedienbar. Als Input sind manchmal tokenisierte, manchmal untokenisierte Textdaten, manchmal bereits mit Wortartentagging vorverarbeitet und manchmal mit Umbrüchen nach jedem Satz (»Ein-Satz-pro-Zeile-Format«) erforderlich. Gute Parser für standardisierte bzw. normalisierte Textdaten sind z. B. der MaltParser (http://www.maltparser.org, Nivre et al. 2006), der Dependenzparser aus den »Mate-Tools« (https://code.google.com/archive/p/mate-tools/, Bohnet 2010), zu dem es sogar eine Online-Demoversion gibt (http://de.sempar.ims.uni-stuttgart.de/) sowie ParZu (https://github.com/rsennrich/parzu, Sennrich et al. 2013), zu dem ebenso eine Onlineversion für einzelne Sätze existiert (https://pub.cl.uzh.ch/demo/parzu/).

Datenspeicherformate für Dependenzstrukturen: Das am einfachsten zu lesende Dateiformat ist das sogenannte CoNLL-Format (https://bit.ly/2umE0oy). In ihm können Dependenzstrukturen wie die in Abb. 2.11 abgebildete Struktur gespeichert werden. Siehe Tab. 2.4 für die Repräsentation der in Abb. 2.11 dargestellten Syntaxstruktur im CoNLL-Format.

1	Ich	ich	PPER	2	SUBJ
2	schreibe	schreiben	VVFIN	0	root
3	dir	du	PPER	2	OBJD
4	morgen	morgen	ADV	2	ADV
5	einen	ein	ART	7	DET
6	ausführlichen	ausführlich	ADJA	7	ATTR
7	Brief	Brief	NN	2	OBJA
8	.	.	$.	0	punct

Tab. 2.4: CoNNL-Datenformat der Struktur aus Abb. 2.11

Im CoNLL-Format erhält jedes Token zunächst einen Index (linke Spalte in Tab. 2.4). Dies geschieht anhand einer fortlaufenden Nummerierung; nach jedem Satz befindet sich eine Leerzeile und die Nummerierung beginnt mit einem neuen Satz von vorne. Allgemein werden in einer CoNLL-Tabelle – neben der bereits besprochenen Annotation von Lemmata und Wortarten in der dritten und vierten Spalte von links – unmittelbare Verweise zwischen den Token eines tokenisierten Texts ausgedrückt (vorletzte Spalte von links), die zusätzlich mit funktionalen Kategorien ausgezeichnet sind (rechte Spalte): In der vorletzten Spalte für die Dependenzbeziehungen wird für jedes Token die zugehörige Mutter spezifiziert. Das hierarchisch höchste Element (die Wurzel) sowie die Satzzeichen (die nicht unmittelbar etwas mit dem syntaktischen Gefüge zu tun haben) erhalten als Verweiswert »0«, was bedeutet, dass sie keinem Element im Satz untergeordnet sind. Jeder Beziehung wird außerdem eine bestimmte syntaktische Funktion zugewiesen (rechte Spalte).

Weitere Speicherformate für Dependenz- und Konstituentenstrukturen werden im nachfolgenden Kapitel 2.2.7.6 vorgestellt.

Visualisierungsprogramme für Dependenzen: Hat man Dependenzannotationen erstellt und im CoNLL-Format ~~oder einem anderen geeigneten Format~~ gespeichert, so lassen sich diese Daten ~~natürlich~~ systematisch durchsuchen bzw. auswerten (s. Kap. 3.1.2.24 f.), aber zu Betrachtungszwecken auch visualisieren. Leicht zu installierende Programme mit dem Hauptzweck der Visualisierung sind der »Depency Viewer« (~~http://~~www.cognitivebase.com/DP/DPViewer.html), in welchem man eingelesene Daten auch editieren und wieder herausschreiben kann, »What's Wrong With My NLP?« (https://bit.ly/2FsJjcI), welches darauf abzielt, mehrere Dependenzannotationen miteinander zu vergleichen, oder der »DG Annotator« (http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/), welcher ebenso eine Vergleichsfunktion besitzt, sich allerdings, was das Annotieren bzw. Editieren angeht, nur zum Korrigieren von Wortarten- und Kanten-Tags eignet (das Erstellen oder Verändern von Dependenzkanten selbst ist unmöglich).

Arbeitsaufgaben

Mit den folgenden Teilaufgaben können Sie die Anreicherung von unverarbeiteten Textdaten mit Dependenzparses nachvollziehen.

1. Verarbeiten Sie den unter der Webadresse https://bit.ly/2CvGbum erhältlichen Text mit dem auf das Deutsche ausgelegte Parser ParZu (https://github.com/rsennrich/parzu), indem Sie die online befindliche Demoversion verwenden: https://pub.cl.uzh.ch/demo/parzu/.
   - Kopieren Sie hierzu den Text in das Input-Fenster, wählen Sie das Ausgabeformat »CoNLL« und klicken Sie »SEND«.
   - Kopieren Sie anschließend die erzeugten Daten in eine Textdatei, die Sie unter dem Dateinamen »Parsen.conll« abspeichern. Achten Sie darauf, dass die Datei in der Kodierung »UTF-8« gespeichert wird.

Mit der folgenden Aufgabe können Sie die geparsten Daten visualisieren.

2. Lesen Sie die Datei in das Programm »DG Annotator« (https://bit.ly/2FnDoEx) ein (nach der Installation des Programms muss man die Datei »dga.jar« ausführen; eine aktuelle Java-Installation ist erforderlich: https://java.com/de/download/).
   - Starten Sie nun das Programm und wählen Sie die Einstellung »Configure« > »Corpus...« > »German«.
   - Laden Sie nun die Datei »Parsen.conll«. Sie können diese auch unter der Webadresse https://bit.ly/2TNFzuQ beziehen. Hinweis: Wenn Sie im »Öffnen«-Dialog des DGA-Annotators den Dateityp »XML, CoNLL file« auswählen, werden nur die verfügbaren Dateien mit der relevanten Dateiendung angezeigt.
   - Sie können nun die analysierten Sätze einzeln betrachten.

Zum Visualisieren und Korrigieren der Daten in Kim Gerdes' ›Arborator‹ s. Aufgabe 3.

3. Öffnen Sie die Webseite https://arborator.ilpga.fr/q.cgi in Google Chrome.
   - Kopieren Sie die Daten aus der Datei »Parsen.conll« in das große Fenster rechts (löschen Sie vorher die dort befindlichen Beispieldaten).
   - Sie können sich die Parses nun anschauen. Um sie mit dem passenden Tagset für syntaktische Funktionen und Wortarten bearbeiten zu können, beziehen Sie die Tags aus der Datei, die Sie unter der Webadresse https://bit.ly/2Yc54oh erhalten. Kopieren Sie die Funktionstags in die Box »additional functions« in Arborator und kopieren Sie die STTS-Wortartentags in die Box »additional POS tags«.
   - Nun können Sie die Daten mit den für sie vorgesehenen Kategorien editieren. Die automatisch erzeugten Parses sind so akkurat, dass es nur wenig zu korrigieren gibt (ein Präpositionalobjekt wurde nicht erkannt, die Anbindung des *dass*-Satzes ist nicht korrekt und ggf. gehört *schon* zu *lange*).
   - Kopieren Sie das Ergebnis der Korrektur (die Tabellendaten auf der rechten Seite) in eine Textdatei namens »Dep_Parse_korrigiert.txt«.

2.2.7.6 | Syntaktische Bäume: Konstituentenstrukturen

Konstituentenstrukturen bilden, wie der Name bereits sagt, ab, welche Elemente im Satz Konstituenten und somit komplexere Strukturen bilden. Konstituenten sind erst einmal nichts Spezifischeres als syntaktische Bausteine von mindestens einem Wort mit einer einheitlichen Funktion im Satz.

Begriffsunterschiede zwischen ›Phrase‹ und ›Konstituente‹: Beide Begriffe sagen aus, dass zusammengehörige Elemente im Satz gruppiert werden, weil sie gewisse Merkmale teilen (z. B. können sie eine gemeinsame, einheitliche Verschiebbarkeit besitzen oder semantische Eigenschaften teilen). Phrasen lassen sich als spezifische Konstituenten beschreiben. So sind Phrasen im Allgemeinen auf syntaktische Komponenten, nicht aber auf morphologische festgelegt, was für den Konstituentenbegriff nicht gilt. Häufig werden die Bezeichnungen ›Konstituentenstrukturbaum‹ und ›Phrasenstrukturbaum‹ jedoch wie Synonyme behandelt.

Zwei Schemata zur Erstellung syntaktischer Konstituenten- bzw. Phrasenstrukturen sind das TIGER-Annotationsschema (https://bit.ly/2HyEv7J, Albert et al. 2003) und das TüBa-D/Z-Stylebook (https://bit.ly/2Y8cSYh, Telljohann et al. 2017), die mit den beiden wichtigsten Baumbanken (dem TIGER- und dem TüBa-D/Z-Korpus) entstanden sind.

Vergleichen Sie die Darstellungen des Beispielsatzes im TIGER- (s. Abb. 2.12) sowie TüBa-D/Z-Annotationsformat (s. Abb. 2.13).

Man sieht, dass die beiden syntaktischen Beschreibungsformate verschiedene syntaktische Aspekte beinhalten. Während das TIGER-Format (Abb. 2.12) sehr flache Phrasenstrukturen vorsieht, beinhaltet das TüBa-D/Z-Format (Abb. 2.13) zusätzlich im syntaktischen Baum topologische Informationen in Form von Klammern und Feldern (Vorfeld = VF, linke Satzklammer = LK, Mittelfeld = MF). Außerdem sieht das TüBa-D/Z-Schema für bestimmte phrasenbildende Wörter wie Adjektive und Adverbien generell Phrasenknoten vor, während das TIGER-Schema dies nur zulässt, wenn die entsprechenden Wörter erweitert werden. Das Inventar an Funktionslabeln sieht bei beiden Schemata syntaktische Funktionen wie Subjekt und Objekttypen vor. Eine Zusammenfassung der Phrasen- und Kantenkategorien, die im TIGER-Korpus vergeben wurden, erhalten Sie

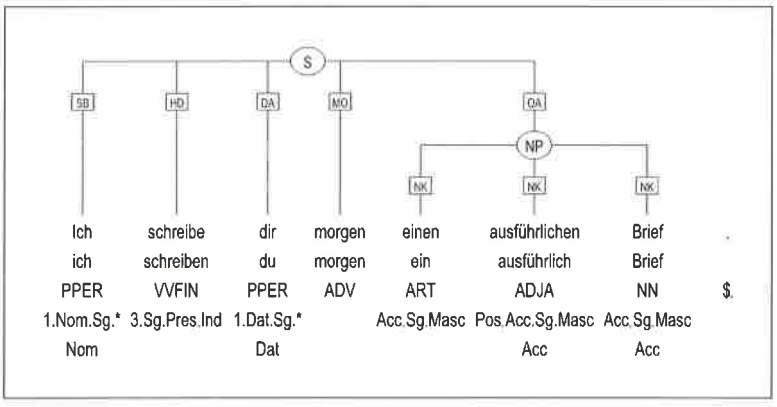


Abb. 2.12: Der Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief* im TIGER-Annotationsformat

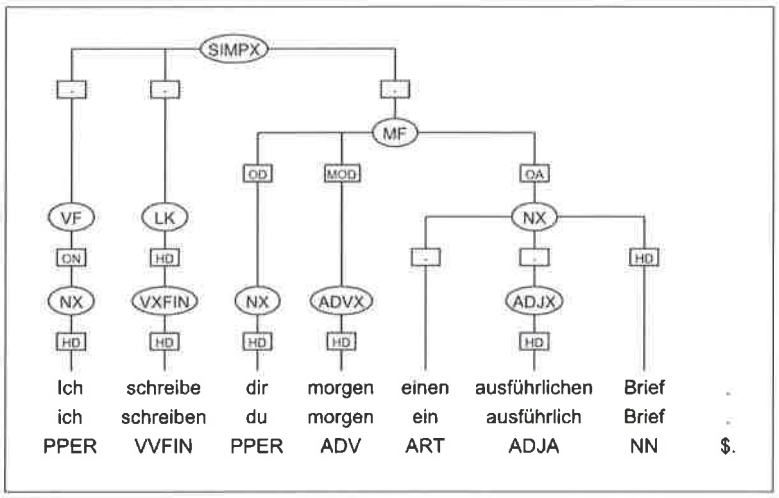


Abb. 2.13: Der Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* im TüBa-D/Z-Annotationsformat

unter der Webadresse https://bit.ly/2HKzyYM. Dieselben Informationen zum TüBa-D/Z-Korpus erhalten Sie unter https://bit.ly/2FqZveh.

Manuelle und automatische Annotation von Konstituentenstrukturen: Für die Annotation von Konstituentenstrukturen existiert derzeit leider kein einfach zu installierendes und zu bedienendes Werkzeug, weil das in der Vergangenheit verwendete Programm »Annotate« (https://bit.ly/2HyG1GS) auf den aktuellen Betriebssystemen nicht mehr unterstützt wird und bislang kein adäquater Ersatz geschaffen wurde. Man kann jedoch automatische Parses von beliebigen Textdaten erstellen. Programme, die dies bewerkstelligen, sind z. B. der Berkeley-Parser (https://github.com/slavpetrov/berkeleyparser) und der Stanford-Parser (https://nlp.stanford.edu/software/lex-parser.shtml), die neben einem Parser des IMS Stuttgart (https://bit.ly/2OphuF3) in dem NLP-Verarbeitungsportal WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblicht/) verfügbar sind. Bis auf WebLicht (zur Bedienung s. Kap. 2.3) lassen sich diese Ressourcen nicht ohne weiterreichende computerlinguistische Kenntnisse bedienen und entsprechende Anleitungen würden den Rahmen dieses Buchs sprengen.

Datenspeicherformate von Konstituentenstrukturen: Die großen deutschen Baumbanken im Konstituentenstrukturformat sind in einem XML-Format gespeichert, das zu dem TIGER-Korpusprojekt entwickelt wurde und deshalb TIGER-XML heißt (https://bit.ly/2TOKUlH). Das gesamte TIGER-Korpus mit knapp einer Million Token können Sie im TIGER-XML-Format herunterladen (https://bit.ly/2TNga4F) und lokal in ein Suchprogramm importieren und durchsuchen (s. Kap. 3.1.2).

Ein schlankes, relativ gut menschenlesbares Repräsentationsformat für syntaktische Bäume ist ein allgemeines Klammerformat, in dem z. B. die englische PENN-Treebank kodiert ist (http://languagelog.ldc.upenn.edu/myl/PennTreebank1995.pdf). Unser Beispielsatz (*Ich schreibe dir morgen einen ausführlichen Brief.*) in der in Abb. 2.12 gezeigten grafischen Interpretation wird in diesem Klammerformat folgendermaßen kodiert:

(1) (S (PPER-SB *Ich*) (VVFIN-HD *schreibe*) (PPER-DA *dir*) (ADV-MO *morgen*) (NP-OA (ART-NK *einen*) (ADJA-NK *ausführlichen*) (NN-NK *Brief*)) ($. .)

Einige der frei verfügbaren Parser geben ihre Analyseergebnisse in diesem Format aus, man muss jedoch berücksichtigen, dass sich in diesem Format gewisse Strukturen wie kreuzende Kanten bei diskontinuierlichen Phrasen, die im Deutschen relativ häufig vorkommen, nicht abbilden lassen. Dies gilt auch für im TIGER- und TüBa-D/Z-Schema vorgesehene sekundäre Kanten, die z. B. für die Beschreibung von Ellipsen verwendet werden. Die folgenden komplex strukturierten Datenformate (die meisten sind XML-basiert) sind das bereits erwähnte TIGER-XML-Datenformat, das von WebLicht ausgegebene TCF-Format (https://bit.ly/2uiKuox), das für die Prager Dependenzbaumbank verwendete pml-Format (http://ufal.mff.cuni.cz/jazz/PML/) oder das in ANNIS (s. Kap. 3.1.2) verwendete Datenmodell Salt (http://corpus-tools.org/salt/). Sie können Feinheiten wie kreuzende Kanten und sekundäre Kanten abbilden. Unterschiedliche Konverter sind in der Lage, zwischen den verschiedenen Abbildungsformaten hin und her zu konvertieren, so z. B. das Programm Pepper (http://corpus-tools.org/pepper/).

Für eine eigenständige Erzeugung von Konstituentenstrukturparses s. Kapitel 2.3.

2.2.8 | Textlinguistische Kategorien: Anaphern und Diskursreferenz

Gehen wir von syntaktischen weiter zu grammatischen Annotationen oberhalb der Satzebene, so gelangen wir zu textlinguistischen Kategorien und Konzepten. Diese können verschiedene Verweise zwischen verschiedenen Sätzen (z. B. durch textdeiktische Elemente wie das demonstrative *dies*) und anderen informationsstrukturellen bzw. textlinguistischen Größen darstellen. Stede (2007) bietet eine umfassende Zusammenschau von korpusbasierten textlinguistischen Analysen, von denen im Folgenden zwei kurz zusammengefasst werden. Stede und Kollegen haben außerdem ein mit textstrukturellen Kategorien vielseitig annotiertes Korpus erstellt, das Potsdamer Kommentarkorpus PCC, http://angcl.ling.uni-potsdam.de/resources/pcc.html. Die Annotationsrichtlinien und viele technische Details dazu wurden im zum Korpus gehörigen Korpushandbuch (Stede 2016) veröffentlicht.

Bezüge zwischen Elementen in verschiedenen Sätzen sind ein wesentlicher Bestandteil der Textbedeutung. Durch das Aufdecken solcher Bezüge werden gleichermaßen wichtige Bedeutungsträger (Diskursreferenten) und somit thematische Aspekte des Textes aufgedeckt. Die meisten Bezüge zwischen Elementen im Text sind dabei auf den vorangegangenen Kontext gerichtet (anaphorisch) und liegen zwischen den Wörtern vor, die auf dieselben Elemente in der Welt verweisen. Wir sprechen hierbei von Koreferenz und meinen damit, dass zwei oder mehr sprachliche Ausdrücke auf denselben Diskursreferenten verweisen. Diskursreferenten

sind dabei die Entitäten, über die im Text gesprochen wird und deren Nennung sich wie eine Perlenkette durch den Text ziehen kann (vergleiche die unten stehenden Beispiele (1)–(3)).

Prototypischerweise werden Diskursreferenten durch einen indefiniten Ausdruck (z. B. ein Nomen mit unbestimmtem Artikel) eingeführt und dann mit definiten Ausdrücken (Nomina mit bestimmtem Artikel, Eigennamen oder Pronomina) wieder aufgegriffen. In vielen Kontexten können jedoch Diskursreferenten gleich definit eingeführt werden, weil das Auftreten gewisser Referenten, wenn auch dem Leser unbekannt, nicht eingeführt werden muss. Vergleiche dazu die folgenden Sätze aus dem TIGER-Korpus.

(1) ***Detlev Kiel*** *von der zuständigen Verwaltungsstelle der IG Metall beschreibt die Ex-DDR als ein Land des Schweigens* . (TIGER-Korpus, Satz 1827)

(2) » *Die Menschen reden nicht mehr mit der Politik* .« (TIGER-Korpus, Satz 1828)

(3) *Auch bei der Zusammenarbeit mit betrieblichen Interessenvertretern sieht* ***er*** *Grenzen* . (TIGER-Korpus, Satz 1829)

Die Sätze 1827 bis 1829 des Tigerkorpus entstammen einem Kultur-Artikel aus der Tageszeitung »Frankfurter Rundschau«. Im ersten der gezeigten Sätze wird der Diskursreferent *Detlev Kiel* eingeführt und im übernächsten Satz durch das Pronomen *er* wieder aufgenommen. Für den menschlichen Leser ist es nicht schwer, diesen Bezug herzustellen. Ohne menschliches Textverstehen ist es allerdings nicht trivial, den korrekten Bezug herzuleiten, zumal er über einen Satz, in dem der Referent nicht explizit genannt wird, hinausgeht.

Annotation von Koreferenz: Wenn man Diskursreferenz auszeichnen will, muss man für beliebig viele Wörter und Wortgruppen im Korpus eine Markierung vornehmen und somit ausdrücken, dass sie denselben Referenten bezeichnen. Dies wird idealerweise mit sogenannten ›pointing relations‹ erreicht, das sind direkte Verweise zwischen Token oder Spannen von Token. Ein Annotationswerkzeug, das zur manuellen Annotationen solcher Verweise erstellt wurde, ist MMAX2 (http://mmax2.net, https://bit.ly/2JuGETQ, Müller/Strube 2006). Die als zum selben Diskursreferenten gehörigen Einheiten werden jeweils mit derselben farblichen Markierung versehen. Mit dem Annotationsprogramm WebAnno (https://webanno.github.io/webanno/, Eckart de Castilho et al. 2016) können Koreferenzannotationen ebenso erstellt werden. In beiden Programmen werden referenzielle Ausdrücke markiert und als miteinander koreferent annotiert, so dass beliebig vielen Referenten im Text beliebig viele koreferente Ausdrücke zugeordnet werden können. Die Abb. 2.14

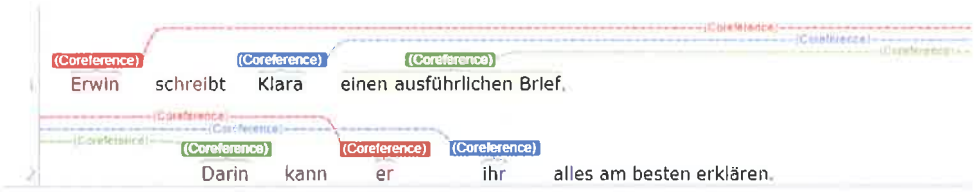


Abb. 2.14: Visualisierung der Satzfolge *Erwin schreibt Klara einen ausführlichen Brief. Darin kann er ihr alles am besten erklären.* mit drei koreferenten Ausdrücken in dem Annotationswerkzeug WebAnno

zeigt eine in WebAnno erstellte Annotation der Satzfolge *Erwin schreibt Klara einen ausführlichen Brief. Darin kann er ihr alles am besten erklären.* mit insgesamt drei koreferenten Ausdrücken.

Das Programm CorZu (https://github.com/dtuggener/CorZu, Tuggener/Klenner 2012) erstellt automatische Koreferenzanalysen und verfügt über eine Online-Demoversion (https://pub.cl.uzh.ch/demo/corzu/, s. u. Arbeitsaufgabe).

Datenspeicherformate: Koreferenzannotationen können in XML-Formaten oder auch in CoNLL-Tabellen gespeichert werden, die bereits im Rahmen von Dependenzannotationen (s. Kap. 2.2.7) vorgestellt wurden. Diese Annotationen können weiterhin mit anderen Speicherformaten zusammengebracht werden, sofern unabhängig voneinander derselbe Text mit derselben Tokenisierung, aber mit verschiedenen Kategorien annotiert wurde. Unter der Voraussetzung identischer Tokenebenen können verschiedene parallele Speicherformate zusammengeführt werden (vergleiche z. B. das Konvertierungsprogramm Pepper, http://corpus-tools.org/pepper/index.html, Zipser/Romary 2010).

Arbeitsaufgabe

- Öffnen Sie in einem Internetbrowser die Seite https://pub.cl.uzh.ch/demo/corzu/ (es handelt sich um die Online-Demoversion des Programms CorZu.
- Geben Sie die Sätze *Erwin schreibt Klara einen ausführlichen Brief. In diesem kann er ihr alles am besten erklären.* als Text in das Eingabefeld ein.
- Betätigen Sie die Einstellung »CoNLL« und bestätigen Sie (»Proceed«).
- Überführen Sie die ausgegebenen Daten nach Arborator (https://arborator.ilpga.fr/q.cgi; s. Arbeitsaufgabe 1 in Kap. 2.2.7.5 für Bedienungshinweise).
- Fügen Sie die fehlende Koreferenzbeziehung (3) hinzu (dies müssen Sie auf der rechten Seite in den Tabellendaten tun) und korrigieren Sie die Dependenzannotationen.
- Speichern Sie das Ergebnis der Bearbeitung (die Tabelleninformationen rechts) in einer Textdatei namens »Koref_korrigiert.txt«.

2.2.9 | Textlinguistische Kategorien: Rhetorische Strukturtheorie (RST)

Die Rhetorische Strukturtheorie (Rhetorical Structure Theory, Mann/Thompson 1988, http://www.sfu.ca/rst/), kurz RST, ist eine Möglichkeit, die Textbedeutung über Analyse elementarer Diskurseinheiten (Englisch: Elementary Discourse Units, kurz EDU) und ihrer Verknüpfungsbedeutung abzubilden. In der Theorie wird angenommen, dass sich die Textbedeu-

tung in einer hierarchischen Struktur mit verzweigenden Knoten wie in der Syntax abbilden lässt. Jeder kohärente Text muss also in Form eines zusammenhängenden Strukturbaums dargestellt werden können, in dem alle EDUs unmittelbar oder mittelbar miteinander verbunden sind.

Das Kernstück der RST sind Annotationsrichtlinien, die zum einen die Segmentierung des Textes in EDUs regeln, zum anderen ein Inventar an Verknüpfungsbedeutungen für die Kanten zwischen einzelnen EDUs anführen (eine Übersicht: http://www.sfu.ca/rst/01intro/) und Tests zur Unterscheidung dieser semantischen Kategorien vorschlagen. Manfred Stede (2016) hat eine aktuelle deutsche Version von RST-Richtlinien veröffentlicht.

Annotation von RST-Strukturen: Die Segmentierung von Texteinheiten in EDUs erfolgt vorwiegend nach dem semantischen Prinzip, EDUs als Einheiten mit Äußerungsstatus zu definieren, unabhängig von ihrer syntaktischen Form. Es werden jedoch auch klare Formvorgaben gemacht, indem bestimmte Phrasen- und Satztypen als EDUs zugelassen werden, andere nicht (z. B. werden subjunktionale Nebensätze generell als EDUs zugelassen, Relativsätze nicht). Nach der Segmentierung müssen die Beziehungen zwischen den EDUs zugeordnet werden. Die meisten Diskursrelationen sind asymmetrisch, das heißt, eine EDU drückt eine für das Gesamtthema des Textes wichtigere Information aus (diese bezeichnet man als »Nukleus«), während die andere EDU untergeordnet ist (diese nennt man »Satellit«). Diese asymmetrischen Diskursrelationen heißen monunuklear. Bestimmt man also die Diskursrelation, die zwischen zwei EDUs besteht, so trifft man auch eine Entscheidung darüber, welche EDU der Nukleus und welche der Satellit ist. Es gibt auch einige Relationen, die die RST als multinuklear bezeichnet. Hier sind die verbundenen EDUs von gleichem Gewicht (und keine ist über- oder untergeordnet). Für jede Verbindung muss aus dem Inventar an Relationen eine Relation gewählt werden (z. B. »Cause« für kausale Relationen, »Antithesis« für adversative, mononukleare Beziehungen).

Nehmen wir für ein Minimalbeispiel für eine RST-Analyse die bereits verwendete Satzfolge:

(1) *Ich schreibe dir morgen einen ausführlichen Brief. Da kann ich dir alles am besten erklären.*

Wir sehen, dass in dem Beispiel kein sprachlicher Ausdruck enthalten ist, der den Zusammenhang der zwei Sätze explizit macht (wie z. B. in der Satzfolge *Ich schreibe dir morgen einen ausführlichen Brief, denn da kann ich dir alles am besten erklären.*). Ein kausaler Zusammenhang liegt jedoch auf der Hand: Der zweite Satz scheint für den Sprecher eine Begründung für die vorangegangene Äußerung zu sein. Nun sieht das RST-Inventar an EDU-Verknüpfungen verschiedene kausale Beziehungen vor: Die Oberkategorie »Cause« mit einigen Unterkategorien beschreibt objektive Kausalzusammenhänge, die sich durch eine Beziehung zwischen einem Verursacher und einem Resultat auszeichnen, wobei das Resultat der Nukleus der Verknüpfung sein muss (vgl. Mann/Thompson 1988, S. 274 f.; z. B. sind die Teilsätze in *Die Tür ist geschlossen, weil der Wind*

MARG.: "Was bestimmt die Größe von EDUs?"

MARG.: "Eine EDU ist Nukleus oder Satellit" / Jeweils einfache Aufz.

MARG.: "EDUs werden über bestimmte Diskursrelationen miteinander verbunden"

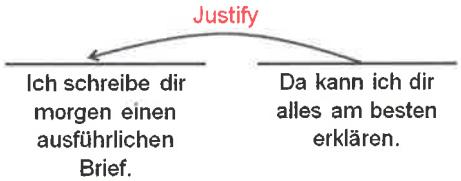


Abb. 2.15: RST-Analyse der Satzfolge *Ich schreibe dir morgen einen ausführlichen Brief. Da kann ich dir alles am besten erklären.*

sie zugeschlagen hat. durch eine Kausalbeziehung verknüpft). Davon werden »Justify«-Beziehungen abgegrenzt, bei denen keine Ursache-Resultat-Beziehung besteht, sondern eine Rechtfertigungsbeziehung (vgl. Mann/Thompson 1988, S. 252 f.). Dies gilt für die beispielhafte Satzfolge in (1): Das Schreiben des Briefes wird nicht bedingt durch einen Erklärvorgang (den zweiten Satz), sondern durch den zweiten Satz gerechtfertigt. Deshalb gilt nach Mann und Thompson (1988) für das Beispiel in (1) die in Abb. 2.15 visualisierte Interpretation.

Als Annotationswerkzeug speziell für RST-Strukturen wurde das »RSTTool« entwickelt (http://www.wagsoft.com/RSTTool/, O'Donnell 1997), dessen Funktionalität in einem allgemeineren Textanalysetool »UAM Corpus Tool« (http://corpustool.com/index.html) implementiert wurde. »RstWeb« ist ebenso ein Annotationswerkzeug für RST-Strukturen (https://corpling.uis.georgetown.edu/rstweb/info/, Zeldes 2016). Eine Zusammenfassung der RST-relevanten Annotationsressourcen finden Sie auch auf einer entsprechenden Unterseite zur RST-Theorie: http://www.sfu.ca/rst/06tools/.

Wenn Sie einen vollständigen Text nach RST-Richtlinien annotieren, werden Sie wahrscheinlich feststellen, dass häufig verschiedene Lesarten im Sinne der RST-Strukturen möglich sind und deshalb ein Festlegen auf eine konkrete Analyse nicht leichtfällt. Dies liegt daran, dass RST-Analysen, von den konkreten sprachlichen Ausdrücken abstrahierend, rekonstruieren, was die Intention eines Textes ist, oder zumindest, wie der Zusammenhang zwischen Äußerungen, der häufig nicht explizit gemacht wird, zu deuten ist.

Allgemein gilt für das Annotieren semantischer Kategorien, dass gegenüber formbasierten und morphosyntaktischen Kategorien ein relativ hoher Interpretationsspielraum gegeben ist. Dies führt zum einen dazu, dass verschiedene Annotatoren häufig nicht zu derselben Analyse gelangen (s. auch Kap. 2.5.2), zum anderen ist es kaum möglich, für Annotationen mit hohem Interpretationsspielraum automatische Werkzeuge zu erstellen, weil ihre Fehlerrate, wenn diese überhaupt bestimmbar ist, oberhalb jeglicher Toleranzgrenze liegt. Deshalb müssen RST-Analysen bislang grundsätzlich per Hand erstellt werden; lediglich bei der Segmentierung der Textdaten helfen manche RST-Programme den Annotatoren.

Die folgenden Arbeitsanweisungen helfen Ihnen bei der Erstellung von RST-Analysen mittels RSTTool.

Anleitung

- Laden sie von der Webseite https://bit.ly/2FijA5o das Programm RSTTool auf ihren lokalen Computer und installieren Sie es gemäß den Anweisungen auf der Homepage.
- Öffnen Sie das Programm und wählen Sie »File« > »Import Text« und laden Sie den Text aus der Datei, die Sie unter der Webadresse https://bit.ly/2CGj54H beziehen können.
- Wählen Sie unter dem Menüpunkt »Text« die Einstellung »Paragraphs«. Die drei elementaren Diskurseinheiten werden anhand der Textumbrüche segmentiert.

- Wählen Sie die Funktion »Relations« > »Load«. Wählen Sie aus dem Ordner »Relation-Sets« im Programmordner des RSTTools die Datei »ClassicMT.rel«, welche das Standardset an RST-Relationen beinhaltet.
- Wählen Sie »Structurer« und dort die Funktion »Link«. Halten Sie den rechten Satz mit der linken Maustaste und lassen Sie über dem mittleren Satz los. Wählen Sie die Relation »Volitional Cause«.
- Klicken Sie auf »Add Span« und klicken Sie auf den mittleren Satz, um einen gemeinsamen Anknüpfungspunkt für den zweiten und dritten Satz zu schaffen.
- Verknüpfen Sie den ersten Satz mit der Schnittstelle 2–3 und bezeichnen Sie die Verknüpfung als »Background« (es muss die Einstellung »Link« gewählt sein).
- Speichern Sie das Ergebnis als »RST_Loesung1.rs3«.

Arbeitsaufgabe

- Führen Sie die Analyse noch einmal durch, so dass der zweite Satz (mit dem dritten Satz zusammen) Satelliten für den ersten Satz mit der Funktion »Elaboration« sind.
- Speichern Sie das Ergebnis als »RST_Loesung2.rs3«.

2.3 | WebLicht: eine Online-Plattform zur automatischen Verarbeitung von Korpusdaten

Viele der in den vergangenen Kapiteln vorgestellten Annotationen können auf dem im Internet zugänglichen Portal WebLicht (https://weblicht.sfs.uni-tuebingen.de/weblicht/) automatisch und aneinandergereiht durchgeführt werden, so dass als Ergebnis eigens eingelesene Textdaten mit komplexen Annotationen versehen wieder ausgegeben werden können. Die Verarbeitungskette lässt sich individuell gestalten, sofern man den fortgeschrittenen Modus nutzt.

Zusammenfassung der Analysemöglichkeiten: Für einen beliebigen deutschen Eingabetext kann man folgende Verarbeitungsschritte auf dem WebLicht-Portal durchführen: Tokenisieren; Normalisieren (Vorschläge für nicht normgerechte Schreibungen in eine gesonderte Annotationsebene schreiben lassen); Wortarten nach STTS oder alternativen Kategorisierungen ausgeben; Lemmata ausgeben; Eigennamen ausgeben; Satzspannen generieren; Phrasenkategorien in Form von Chunks ausgeben; Phrasenstrukturbäume erstellen; Dependenzstrukturbäume erstellen; ein Lexikon indizieren; Frequenzinformationen zu Wortformen abrufen.

Das Ausgabeformat für die meisten in WebLicht erzeugten Analysen ist das XML-basierte TCF-Format (https://bit.ly/2uiKuox), das in andere

Formate umgewandelt werden kann. Im Regelfall kann man sich die in WebLicht erzeugten Analysen auch unmittelbar visualisieren lassen.

Die folgenden Arbeitsaufgaben stellen eine standardmäßige Verarbeitungskette in WebLicht dar, die einen Parsingprozess mit Konstituentenstrukturen beinhaltet.

Anleitung

- Loggen Sie sich unter der Webseite https://weblicht.sfs.uni-tuebingen.de/weblicht/ ein. Sie benötigen hierfür einen Account bei einer wissenschaftlichen Einrichtung. Klicken Sie »Start«.
- Wählen Sie »Browse« und importieren Sie den Beispieltext, den Sie unter der Webadresse https://bit.ly/2CvGbum beziehen können. Wählen Sie dann die Sprache »German« und klicken Sie »OK«.
- Wählen Sie »Advanced Mode«.
- Ziehen Sie die folgenden Anwendungen (per ›Drag & Drop‹) nacheinander aus dem oberen Bereich in die untere Verarbeitungskette: »SfS: To TCF Converter«, »IMS: Tokenizer«, »IMS: TreeTagger« und »IMS: Constituent Parser«.
- Wählen Sie »Run Tools«. Warten Sie, bis die Verarbeitungskette durchlaufen ist. Sie können anschließend die Analyseergebnisse zum Weiterverarbeiten herunterladen, aber auch gleich online betrachten: Klicken Sie bei dem »IMS: Constituent Parser« auf »Visualize Result« und im Ergebnisfenster auf »parsing«. Sie können nun die syntaktische Analyse des Parsers betrachten.
  Das Ergebnis des Werkzeugs »IMS: TreeTagger« können Sie z. B. herunterladen und im EXMARaLDA-Partitureditor als »TCF file« importieren. So lassen sich größere Datenmengen per Knopfdruck vorverarbeiten und in EXMARaLDA weiterverarbeiten.
  Sie können den zu verarbeitenden Text sowie die in der Verarbeitungskette befindlichen Werkzeuge frei variieren.

2.4 | Vom gesprochenen Text zum Korpus: Erstellung von Gesprächskorpora

Definition

Nach Deppermann/Schmidt (2014) bezeichnet der Begriff **Gesprächskorpus** Korpusdaten, die auf der Grundlage von mündlichen oder gestikulierten Äußerungen und entsprechend Audio- oder Videodateien erstellt wurden.

Die korpuslinguistische Aufbereitung medial mündlicher Sprache entspricht zu großen Teilen der bereits behandelten Aufbereitung medial schriftlicher Sprache. Zwei wesentliche Unterschiede bzw. Ergänzungen können wir dabei jedoch feststellen:

1. Während schriftliche Daten bereits im Zielmedium vorliegen, müssen mündliche Daten zunächst in das schriftliche Medium überführt werden. Diesen Prozess nennt man Transkription bzw. Transkribieren. Man benötigt diesen Prozess, weil ein Zuweisen von linguistischen Analysen an ein akustisches Signal nur sehr bedingt möglich ist und weil die Verschriftlichung des akustischen Signals einen notwendigen Abstraktionsprozess bedeutet, ohne den jede akustische Sequenz eine idiosynkratische Stelle im Korpus wäre (denn selbst wenn wir inhaltlich exakt dieselben Dinge mehrfach äußern, entstehen dabei akustisch jeweils verschiedene Signale).
   Die Transkription der gesprochensprachlichen Daten kann nach verschiedenen Richtlinien erfolgen, gemäß deren das Sprachsignal unterschiedlich stark normalisiert wird. Darauf muss der anschließende Normalisierungsprozess abgestimmt sein: Um eine normalisierte Verschriftlichung zu erhalten, muss stärker normalisiert werden, je näher sich die Transkription am Audiosignal orientiert.
2. Bezogen auf den mündlichen Sprachgebrauch interessieren sich Linguistinnen und Linguisten neben all den bereits erwähnten grammatischen Aspekten zusätzlich für im Mündlichen auftretende Phänomene wie eine dialogische Textstruktur mit teilweise simultan ablaufenden Redeanteilen, bestimmte Merkmale des Audiosignals (z. B. Tonhöhe und Sprechgeschwindigkeit) oder Performanzphänomene wie Pausen, bestimmte Korrekturen, Wiederholungen usw., die häufig unter dem Begriff Häsitationsphänomene zusammengefasst werden (s. Kap. 2.4.4–2.4.9, vgl. Imo/Lanwer 2019).

2.4.1 | Transkribieren

Definition

Transkribieren bezeichnet den Prozess des Überführens medial mündlicher (also auditiver) und gestisch kommunizierter Sprachdaten in eine schriftliche Form. Das **Transkript** oder auch die **Transkription** ist das Resultat dieses Prozesses. Es bzw. sie dient der weitergehenden linguistischen Analyse des ursprünglichen Kommunikationssignals, stellt aber auch selber bereits eine linguistische Analyse dar.

In einem weitergefassten Begriffsrahmen kann Transkribieren allgemein das Überführen von Sprachdaten in eine leichter interpretierbare Form bezeichnen und auch auf schriftlich kommunizierte Sprache angewendet werden. Zum Beispiel können schwer menschen- und computerlesbare Handschriften durch Transkriptionen besser zugänglich gemacht werden.

Transkription und Annotation: Bei der Definition des Annotationsbegriffs (s. Kap. 2.2.1) wurde bereits darauf hingewiesen, dass es sich beim Transkribieren um eine bestimmte Art der Annotation handelt: Wie bei allen Annotationen der Fall, verwendet man bestimmte Richtlinien, um eine Interpretation der zugrunde liegenden Daten zu erzielen, die

wiederum in einer meist endlichen (manchmal aber auch unendlichen) Anzahl an Kategorien abgebildet wird.

Bei der Transkription eines Audiosignals kann man versuchen, die Eigenschaften des mündlichen Sprachsignals so detailliert wie möglich abzubilden. Dies erreicht man, indem man ein bestimmtes Transkriptionssystem (s. u.) verwendet und/oder auf Lautschriften wie das Internationale Phonetische Alphabet (IPA, http://www.internationalphoneticalphabet.org/) zurückgreift. So detailgetreu man aber auch vorgeht, es werden immer bestimmte Eigenschaften des akustischen Signals unabgebildet bleiben, womit die Eigenschaften des Audiosignals normalisiert werden. Mit anderen Worten: Transkription ohne Normalisierung ist nicht möglich. Somit müssten prinzipiell dieses und das folgende Kapitel 2.4.10 als ein (mehrstufiger) Prozess zusammengefasst werden. In der Praxis der Korpusaufbereitung werden jedoch zwei Vorgänge getrennt: Einmal entscheiden wir uns für eine bestimmte Transkriptionspraxis mit einem bestimmten Grad an Normalisierung, einmal nehmen wir die verbleibende Anpassung der Transkription an die schriftsprachliche Norm vor. Nur in dem Extremfall, in dem wir mündliche Sprache in einem Schritt in einen Standardtext überführen, vereinen wir beide Verarbeitungsschritte untrennbar.

Das Instrumentarium zur Transkription akustischer Sprachsignale besteht zum einen aus Zeichen zur Repräsentation des eigentlichen sprachlich relevanten Materials, also z. B. die bereits erwähnten IPA-Symbole wie das [ʃ] für die Lautung der im Deutschen <sch> geschriebene Zeichenkette. Gleichzeitig besteht das Bedürfnis, weitere Phänomene wie z. B. die zeitliche Verortung der gesprochenen Wörter im Dialog, Pausen im Sprechfluss usw. abzubilden, vor allem um in der Gesprächsanalyse bestimmte pragmatische Phänomene analysierbar zu machen.

Transkriptionsrichtlinien: Der bekannteste Analyseanasatz für die Transkription deutscher Sprache ist das Gesprächsanalytische Transkriptionssystem GAT (http://www.mediensprache.net/de/medienanalyse/transcription/gat/, Selting et al. 1998) bzw. die weiterentwickelte Version GAT 2 (Selting et al. 2009), das ein Beschreibungsinventar für die folgenden gesprächsspezifischen Phänomene beinhaltet:

- Metadaten (s. Kap. 2.6)
- Zuweisung transkribierter Sprecherbeiträge (sog. Turns) zu Gesprächsteilnehmern
- Überlappung von Äußerungen
- Markierung schneller Turn-Anschlüsse (ohne Pause zwischen den Turns)
- Markierung von Pausen innerhalb und zwischen Turns
- Dehnungen
- gefüllte Pausen
- Glottalverschluss innerhalb von Wörter (Abbruchsignal)
- Rückkopplungssignale: Lachen und Nichtwörter als Rezeptionssignale (*hmmm*)
- Wortakzente
- Tonhöhenverläufe am Turn-Ende
- Kommentierungen nicht-sprachlicher Handlungen der Gesprächsteil-

nehmer (wie seufzen etc.) sowie Kommentare der Transkribenten hinsichtlich der Gemütslage der Gesprächsteilnehmer sowie des transkribierten Materials (Unsicherheiten bei der Interpretation)

In Selting et al. (2009), S. 394, findet sich ein Transkriptionsbeispiel, von dem in Beispiel (1) die ersten 13 Zeilen abgebildet sind.

(1)

```
01  S1:  ja:; (.) die VIERziger generation so;=
02       =das_s: !WA:HN!sinnig viele die sich da ham
         [SCHEI]den
03  S2:  [ja;  ]
         lasse[n.=]
04  S2:       [hm,]
05  S1:  =oder scheiden lassen Überhaupt.
06  S2:  hm,
07       (--)
08  S1:  heute noch-
09       ((atmet 2.1 Sek. aus))
10       s_is der UMbruch.
11  S2:  n besonders GUtes beispiel das warn mal unsere
         NACHbarn.
12       (---)
13       ähm    (---)
```

Die linke Nummerierung ist keine Zeilennummerierung, sondern die Durchnummerierung der einzelnen Redebeiträge (Turns). In der Spalte rechts daneben finden sich die Bezeichnungen für die verschiedenen Sprecherinnen und Sprecher. Wiederum rechts daneben findet sich die eigentliche Transkription, bestehend aus einer mit dem deutschen Alphabet verfassten sogenannten literarischen Umschrift und zusätzlichen Informationen zum gesprochenen Text gemäß den oben aufgeführten Möglichkeiten. Die literarische Umschrift lässt Abweichungen von der Standardorthographie zu, sofern noch erkennbar ist, um welches Wort es sich handelt (vgl. Selting et al. 2009, S. 360). Für alle nicht artikulatorisch markierten Wörter soll allerdings die deutsche Standardorthographie verwendet werden. Da die Großschreibung jedoch zur Markierung von Wortakzenten dient, gelten nicht die Regeln der deutschen Großschreibung. Die Symbole um und in den Wortformen bedeuten konkret (in der Reihenfolge ihres Auftretens im Transkript):

:	Dehnungszeichen für (über ein Normalmaß hinausgehende) Laute
;	fallende Intonation
(.)	Mikropause
=	Turnübernahmesignal
[]	Bereich überlappender Redeanteile
,	steigende Intonation
(--)	mittlere geschätzte Pause von ca. 0.5–0.8 Sek. Dauer
-	gleich bleibende Intonation

MARG: "Bedeutung der in (1) auftretenden Transkriptionssymbole"

(( )) Kommentar zum Sprecherverhalten
_ Markierung einer klitisierten Einheit
(---) längere geschätzte Pause von ca. 0.8–1.0 Sek. Dauer

GAT-Transkriptionen, wie beschrieben in Selting et al. 1998 und 2009, sind nicht mit der modernen Korpusmethodik vereinbar, weil ein Transkript wie das oben gezeigte nicht zur Erstellung eines modernen Korpus führen kann. Die Gründe dafür sind vielzählig: Vor allem bieten GAT-Transkripte wie das obige keine einheitliche Tokenisierung und lassen aufgrund der Annotation sämtlicher Informationen auf ein und derselben Beschreibungsebene keine Normalisierung zu, durch die eine systematische Suche von Wortformen gewährleistet wird. Die verschiedenen Beschreibungsebenen (Artikulation, Prosodie, Pausen usw.) sind nicht klar auseinanderzuhalten, sondern die Beschreibungsebene kann sich mit jedem Zeichenabstand ändern. Dieses Zusammenmischen aller Informationen auf einer Annotationsebene führt zu kontraintuitiven Entscheidungen bei der GAT-Erstellung. Zum Beispiel dürfen Wörter generell nicht mit Bindestrichen geschrieben werden, weil der Bindestrich für die Markierung der Intonation verwendet wird. Dies wäre vermeidbar, wenn man eine Annotationsebene für die Beschreibung der artikulierten Wörter und eine getrennte Annotationsebene für die Intonation verwenden würde – der Bindestrich hätte dann jeweils eine unterschiedliche Bedeutung und die verschiedenen Bedeutungen würden sich nicht vermischen.

Trotz all dieser korpuslinguistischen Ungereimtheiten bietet GAT ein wertvolles Instrumentarium für die Beschreibung gesprochensprachlicher Phänomene. Wir können die GAT-Konventionen korpuslinguistisch nutzen, indem wir die Beschreibungsebenen trennen und die vorgeschlagenen Kategorisierungen und Symbole dann auf diesen Ebenen verwenden. Weil dies der einzige Weg ist, den Transkriptionsprozess mit der modernen korpuslinguistischen Methodologie zu vereinbaren, wird der Transkriptionsprozess auseinandergezogen: Während die GAT-Konventionen versuchen, möglichst viele gesprochensprachliche Merkmale auf einer Beschreibungsebene abzubilden, werden diese in den nachfolgenden Kapiteln 2.4.4–2.4.9 als voneinander unabhängige Annotationen behandelt.

2.4.2 | Transkriptionswerkzeuge

Da im korpuslinguistischen Kontext Transkriptionen nicht nur menschenlesbar, sondern auch maschinenlesbar und durch Analyseprogramme weiterverarbeitbar sein sollen, bedarf es entsprechender Programme. Häufig verwendete Transkriptionsprogramme sind:

- EXMARaLDA (http://www.exmaralda.org)
- ELAN (https://tla.mpi.nl/tools/tla-tools/elan/)
- PRAAT (http://www.fon.hum.uva.nl/praat/)
- Transcriber (http://trans.sourceforge.net/en/presentation.php)
- OCTRA (http://www.phonetik.uni-muenchen.de/apps/octra/octra/features)
- Anvil (http://www.anvil-software.org/)

Die Verwendung dieser oder anderer Ressourcen sollte von dem Verarbeitungsziel bzw. Analyseziel abhängig gemacht werden. Die folgenden Fragen können ausschlaggebend für die Entscheidung sein.

- Soll mit dem Programm aufgenommen (dies spricht z. B. für PRAAT) oder sollen bereits aufgenommene Daten verarbeitet werden?
- Ist das Ausgangssignal monologisch oder dialogisch (EXMARaLDA und ELAN sind z. B. primär für die Darstellung von Dialogizität konzipiert worden)?
- Liegt das Ausgangssignal als Audio- oder Videodatei vor (Anvil ist z. B. spezifisch für Videoaufnahmen, PRAAT für Audioaufnahmen)?
- Soll das Audiosignal auf phonetische Signaleigenschaften hin analysiert werden (PRAAT ist z. B. für die phonetische Analyse konzipiert, EXMARaLDA viel eher für die gesprächsanalytische)?
- Sind grammatische Annotationen, wie in Kap. 2.2.6 erläutert, geplant, die auf der Transkription aufbauen (Mehrebenenannotationen mit verschiedenen Bezügen zwischen den Ebenen bieten sich nur in EXMARaLDA und ELAN an)?
- Welche Datenformate werden von dem jeweiligen Werkzeug erzeugt, welche werden anschließend benötigt (EXMARaLDA verfügt über die vielseitigsten Import- und Exportmöglichkeiten)?

2.4.3 | Anlegen von Transkriptionsprojekten im EXMARaLDA-Partitureditor

In Kapitel 2.2.2 wurden bereits grundlegende Funktionen des EXMARaLDA-Partitureditors behandelt, die sich auf die Erstellung von Annotationen zu schriftlichen Sprachdaten beziehen. Diese sind genauso relevant für mündliche Sprache, man muss durch den Transkriptionsprozess nur zuerst eine Version der Daten erzielen, auf die die vorgestellten Verfahren anwendbar sind.

Ausgehend von einer zugrunde liegenden Audiodatei, sollen die wesentlichen Arbeitsschritte zur Vorbereitung eines Transkriptionsprojekts in der folgenden Schnellanleitung vorgestellt werden.

Anleitung

- Legen Sie einen Arbeitsordner für eine Tondatei und deren Bearbeitung mittels EXMARaLDA auf Ihrem Computer an. Laden Sie in dieses Verzeichnis eine sehr kurze Audioaufnahme, die Sie unter https://hu.berlin/gat_bsp/ herunterladen können. Die Originaldatei zur Vermittlung des Gesprächsanalytischen Transkriptionssystems GAT wird mitsamt einer Beschreibung von GAT bzw. den Transkriptionsrichtlinien unter der Adresse https://bit.ly/2und7Ro bereitgestellt.
- Starten Sie den EXMARaLDA-Partitureditor, ggf. mit Zuhilfenahme der Informationen aus Kap. 2.2.2 oder der EXMARaLDA-Webseite https://www.exmaralda.org.
- Um ein neues Projekt samt zu transkribierenden Primärdaten anzulegen, wählen Sie im obersten Menü die Option »File« > »New from

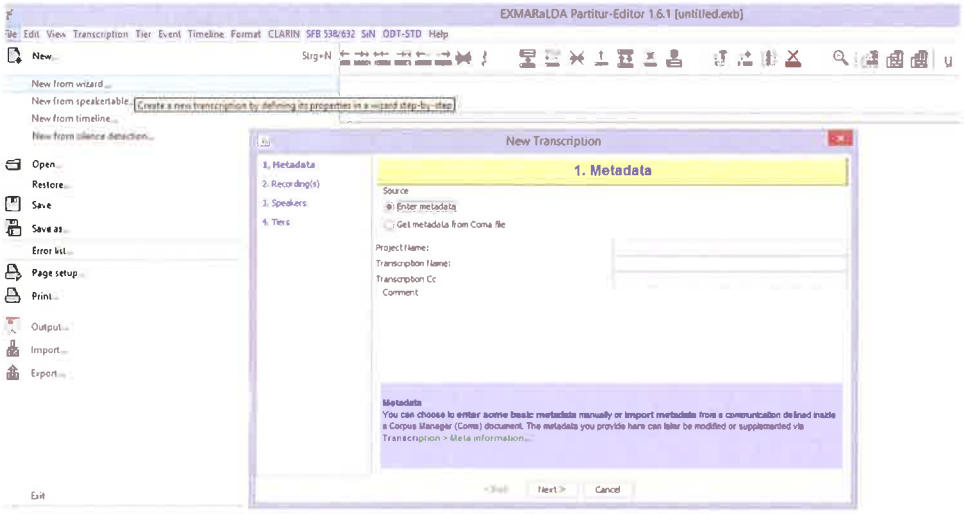


Abb. 2.16: Erste Schritte zum Einrichten eines neuen Transkriptionsprojekts im EXMARaLDA-Partitureditor

wizard...« und navigieren Sie sich wie folgt durch die Optionen zum Anlegen eines neuen Projektes (s. auch Abb. 2.16).

- Geben Sie einen Projektnamen (z. B. »Transkription lernen«), einen Transkriptionsnamen (z. B. »Testtranskription«) und ggf. Transkriptionskonventionen (z. B. »GAT«) ein und klicken Sie weiter.
- Geben Sie den Speicherort der zu transkribierenden Datei (»gat2_beispiel_Anfang.wav«) an.
- Da in der Audiodatei zwei unbekannte Sprecherinnen zu hören sind, spezifizieren Sie zwei Sprecherinnen, indem Sie z. B. »S1,S2« in das Textfeld eingeben.
- Im nächsten Dialogfenster werden die Benennungen der einzelnen Tiers (Korpusebenen) festgelegt. Wie in Kap. 2.2.2 beschrieben, können Sie diese umbenennen oder weitere Annotationsebenen hinzufügen. Praktisch ist, dass man hier Ebenen anlegen kann, die jeweils für die einzelnen Sprecherinnen und Sprecher gelten. Mit Blick auf die in Kap. 2.4.4 f. folgenden einzelnen Schritte zur Transkription von Audiomaterial werden folgende Tiers bzw. Ebenen benötigt:
  - Artikulierte Laute: Die artikulierten Lautabfolgen der beiden Sprecherinnen. Dies sollen die wesentlichen Referenzebenen sein, die in der Hierarchie der Ebenen ganz oben stehen. Diese Ebenen werden im Textfeld »Category for main transkription tier:« eingetragen. Sie können z. B. »TXT« oder »Umschrift« heißen. Diese Hauptebene wird vom Tier-Typ her als Transkription ausgewiesen, wohingegen alle folgenden Ebenen als Annotationen interpretiert werden. Dies ist vielleicht konzeptionell nicht ganz korrekt, basiert aber auf einer technischen Vorgabe, dass nur eine Transkriptionsebene pro Sprecherin existieren soll. Da oberhalb dieser Haupttranskriptionsebene in unserem Fall keine Korpusebene vorgesehen ist, werden die verschiedenen Annotationsebenen unterhalb »...after the main transcription tier« hinzugefügt, indem jeweils auf das »+«-Symbol geklickt wird.

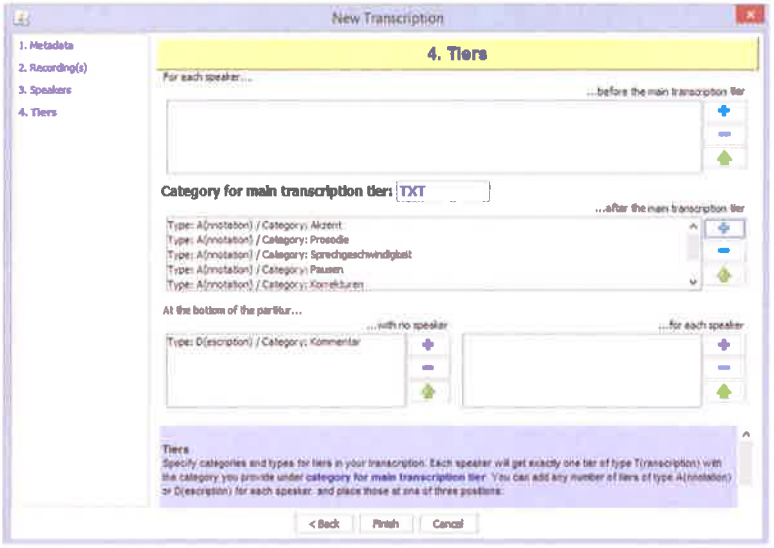


Abb. 2.17: Eingabefenster für sprecherbezogene und sprecherunabhängige Annotationsebenen in EXMARaLDA

- Akzentstrukturen: Die Bezeichnung der Ebene kann z. B. »Akzent« lauten.
- Prosodie
- Sprechgeschwindigkeit
- Pausen
- Korrekturen
- Außersprachliche Geräusche: Da man häufig lange Namen für Annotationsebenen sowie Leerzeichen im Ebenennamen vermeidet, kann die Bezeichnung z. B. »Geräusch« lauten.
- Normalisierung: Die Normalisierungsebene wird häufig mit der Ebenenbezeichnung »norm« versehen.
- Annotatorenkommentare: Diese Ebene ist die einzige, die als sprecherunabhängig eingetragen wird (links-unterer Bereich im Eingabefenster, bei »At the bottom of the partitur...« > »...with no speaker«).

Abb. 2.18: Resultierende EXMARaLDA-Partitur

- Das Eingabefenster sollte nun wie in Abb. 2.17 aussehen.
- Nachdem Sie diese Einstellungen abgeschickt haben, sollte die resultierende EXMARaLDA-Partitur wie in Abb. 2.18 aussehen.

Auf dieser Arbeitsgrundlage können nun die einzelnen Transkriptionsschritte erfolgen. Grundlegende Funktionen bezogen auf die eingelesene Audiodatei und die Anordnung der Transkriptions- bzw. Annotationsebenen sind:

- Das Abhören des Audiosignals: Mit den Abspielfunktionen oberhalb des hellgrauen, leeren Bereichs im Editor und der Markierung von Ab-

spielbereichen im Oszillogramm, welches das Audiosignal grafisch darstellt, kann man die eingelesene Audiodatei im Gesamten und in Abschnitten abhören.
- Die Anordnung der Tiers (zur besseren Interpretierbarkeit der Beschreibungsebenen) kann mithilfe der Funktion »Move tier upwards« in der oberen Reihe an Symbolen verändert werden. Hier können auch Tiers gelöscht, hinzugefügt, verborgen oder eingefügt werden, wenn sie zuvor in die Zwischenablage kopiert wurden.

2.4.4 | Abbildung lautlicher Merkmale im Minimaltranskript

Definition

> Das **Minimaltranskript** ist laut den GAT-Konventionen die Mindestanforderung an eine Transkription, die im Wesentlichen den Dialogverlauf darstellt und eine literarische Umschrift beinhaltet. Wenige weitere Merkmale wie die Notation von Pausen können im Minimaltranskript angegeben werden.

Die wesentlichen Merkmale des Minimaltranskripts werden im Folgenden zusammengefasst. Bitte lesen Sie begleitend zu den Ausführungen das auf der Webseite http://www.mediensprache.net/de/medienanalyse/transcription/gat/ verlinkte Dokument, welches die GAT-2-Transkriptionsrichtlinien vorstellt.

Lautabfolgen (Minimaltranskript): Die wichtigste Aufgabe bei der Erstellung einer Transkription ist das Verschriftlichen der im Sprachsignal hörbaren Laute bzw. Lautsequenzen. Dies ist optimal möglich durch die Verwendung einer Lautschrift wie dem Internationalen Phonetischen Alphabet (IPA, http://www.internationalphoneticalphabet.org/), weil mittels solcher Schriften eine 1-zu-1-Zuordnung zwischen einem bestimmten Laut und einem entsprechenden Zeichen erfolgt, was beim normalen Schreibalphabet nicht gewährleistet ist. Ohne zusätzliches Wissen kann man im Deutschen z. B. nicht wissen, ob man den Buchstaben <s> stimmhaft (also laut IPA [z]) oder stimmlos (also laut IPA [s]) aussprechen muss. Dennoch verzichten die allermeisten Transkriptionen mit gesprochensprachlicher Datengrundlage auf die Verwendung von Lautschriften, weil das Transkribieren nach IPA sehr zeitaufwändig wäre.

Die verschiedenen Korpusprojekte, in deren Rahmen große Mengen von mündlichen Sprachdaten transkribiert und annotiert wurden und werden, verwenden meistens eine literarische Umschrift mit spezifischen Kriterien und verfügen zusätzlich über eine normalisierte Ebene (s. Kap. 2.4.10), auf der wiederum automatische oder manuelle Annotationen hinzugefügt werden. Aktuelle Beispiele hierfür sind das FOLK-Korpus (http://agd.ids-mannheim.de/folk.shtml) oder das GeWiss-Korpus (https://gewiss.uni-leipzig.de/).

Erstellung einer literarischen Umschrift: Eine Möglichkeit bei der Transkription per Hand ist die Nutzung der Annotationswerkzeuge EXMA-

RaLDA oder FOLKER (http://www.exmaralda.org, Schmidt/Wörner 2014). In beiden Werkzeugen können Sie Audiodateien einlesen und von Grund auf bearbeiten, indem Sie mithilfe der Computertastatur die in dem Audiosignal gelauteten Wörter darstellen. Die Richtlinien, welche für das FOLK-Projekt (dem größten aktuellen Projekt zur Erstellung eines Gesprächskorpus, http://agd.ids-mannheim.de/FOLK_extern.shtml) verwendet werden, sind unter dem Namen »cGAT-Handbuch« frei verfügbar (Schimidt et al. 2015, https://bit.ly/2HCnZnc). Auf Seite 10 f. finden Sie die Hinweise zur schriftlichen Abbildung des gesprochenen Signals, das als Erstellung eines »Minimaltranskripts« interpretiert wird. Das heißt, dass laut cGAT die Minimalanforderung an ein Transkript die schriftliche Abbildung der gelauteten Wörter ist.

Zusammengefasst sind geben die Richtlinien des Minimaltranskripts in cGAT vor, dass die artikulierten Wörter nach der deutschen Standardorthographie repräsentiert werden, sofern die Lautung eines Wortes »nicht signifikant von seiner standardsprachlichen Lautung abweicht« (Schmidt et al. 2015, S. 11). Von dieser standardorthographischen Repräsentation ausgeschlossen sind die Großschreibung (alle Wörter werden unabhängig von ihrem Wortstatus und der Position im Satz kleingeschrieben) und die Interpunktion (es wird keine orthographisch motivierte Interpunktion verwendet, sondern die Verwendung von Punktuationsnotationen ist der Markierung von Pausen vorbehalten).

Somit sind die Schreibkonventionen nach cGAT teilweise orthographisch, teilweise phonetisch motiviert, wobei die folgenden Richtlinien für eine orthographische Transkription sprechen:

- die Worttrennung (sie erfolgt nicht gemäß einer phonetischen Segmentierung, sondern gemäß den orthographischen Regeln zur Wortsegmentierung, die wiederum hauptsächlich syntaktische und morphologische Regeln beinhaltet);
- die Wortschreibung: Bei nicht auffällig individuell gelauteten Wörtern wird die Standardorthographie verwendet, allerdings abzüglich der Substantivgroßschreibung. Bei Standardlautung und sich widersprechender orthographischer Repräsentation »gewinnt« die orthographische Repräsentation: Es soll im Fall von [lʊstɪç] *lustig* transkribiert werden, nicht *lustich*, weil die Lautung mit [ç] als die Standardlautung gilt (Schmidt et al. 2015, S. 15). Auch im Fall von Fremdwörtern wird die Fremdwortschreibung angegeben, nicht die eigentliche Lautung.

Für eine phonetische Transkription spricht:

- die Repräsentation nicht standardsprachlich gelauteter Wörter: Bei »signifikanten Abweichungen« (Schmidt et al. 2015, S. 12 f.) von der Standardorthographie wird die Abweichung nach dem Prinzip »Schreib, wie du es hörst« repräsentiert. Dies gilt insbesondere für dialektbedingte Lautungen. Im Fall der oben erwähnten Repräsentation von *lustig* mit der Lautung [lʊstɪk] wird das Wort als *lustik* transkribiert. Die Abweichung von der Standardschreibung gilt ebenso bei Wortabbrüchen.

In Selting et al. (2009), S. 8 f. finden Sie weitere Beispiele für die Realisierung von Lautsequenzen bzw. phonologischen Wörtern.

Darstellung der Dialogizität und Turninteraktionen: Bei Sprechsituationen mit mehr als einem Sprecher, was in mündlicher Kommunikation der Regelfall ist, sind folgende Merkmale der Sprachsignale relevant und sollten im Korpus strukturell annotiert sein:

- die Zuordnung von Redeanteilen zu einem bestimmten Sprecher bzw. einer bestimmten Sprecherin,
- die Überlappung von Redeanteilen.

Gegenüber der Erstellung konventioneller GAT-Transkripte ergibt sich bei der Erstellung moderner Korpora der Unterschied, dass aufgrund der digitalen Aufbereitung keine Zeilenbrüche erzeugt werden. Bestimmte Zeilen im Korpus repräsentieren spezifische Ebenen, so dass einzelne Sprecherinnen und Sprecher einfach bestimmte Ebenen zugewiesen bekommen. Die Transkription zwei verschiedener Sprecherinnen bzw. Sprecher kann also einfach auf zwei verschiedenen Ebenen im Korpus erfolgen. Vergleichen Sie die Transkriptionszeilen 02 bis 04 aus Selting et al. (2009), S. 394:

(1)

```
02   S1:   =das_s: !WA:HN!sinnig viele die sich da ham
           [SCHEI]den
03   S2:   [ja;  ]
     S1:   lasse[n.=]
04   S2:        [hm,]
```

Aus diesem Ausschnitt geht hervor, dass zwei Personen – S1 und S2 – an dem Dialog beteiligt sind und S2 zweimal an bestimmten Stellen simultan zu S1 spricht. Die Notation drückt aus, dass das positive Rückkopplungswort *ja* genau mit der ersten Silbe des Worts *scheiden* überlappt und das Rückkopplungssignal *hm* mit dem letzten Konsonanten des Worts *lassen*. Übertragen wir die Darstellung in eine EXMARaLDA-Partitur mit festen Ebenen für die einzelnen Sprecher, so kommen wir zu der in Abb. 2.19 ersichtlichen Darstellung.

Man kann also auf die eckigen Klammern verzichten, die den Bereich der Überlappung markieren. Mithilfe der Elemente auf der Zeitleiste bzw. Referenzebene kann man sehr genau Segmentgrenzen festlegen und somit beliebige Überlappungen zwischen Zeichensequenzen verschiedener Transkriptionsebenen kenntlich machen.

2.4.5 | Segmentierung der Transkription

Abb. 2.19 ~~im vorigen Kapitel~~ weist eine wortbasierte Tokenisierung auf, d. h. die Segmentierung der Korpusdaten bzw. der Transkriptionsebene erfolgt wortweise. Transkripte werden in den meisten Fällen jedoch nicht nach einer korpuslinguistischen Methodik erstellt, sondern quasi-analog-

Abb. 2.19: EXAMARaLDA-Darstellung des GAT-Transkripts (Zeile 02–04), vgl. Beispiel (1)

S1	=das_s:	!WA:HN!sinnig	viele	die	sich	da	ham	SCHEI	den	lassen
S2								ja;		hm

ohne dass im erstellten Transkript anschließend systematisch bzw. elektronisch nach bestimmten Variablen und Werten gesucht werden kann. Häufig wird primär eine Segmentierung auf Äußerungsebene angestrebt, wobei verschiedene Kriterien angesetzt werden können, z. B. syntaktische, semantische oder prosodische. In Selting et al. (2009), S. 11 f. sowie S. 18 f., wird ausgeführt, dass im Normalfall Intonationsphrasen segmentiert werden – prosodisch als abgeschlossene Einheiten analysierte Wortsequenzen mit einem typischem Tonhöhenverlauf und weiteren prosodischen Charakteristika.

Welche linguistische Kategorie als geeignetes Pendant zum syntaktischen, standardsprachlichen Satz gelten soll, ist derzeit umstritten (vgl. z. B. das Projekt »SegCor« des Instituts für Deutsche Sprache, https://bit.ly/2OfSO1B, oder Auer 2010). Eine korpuslinguistisch brauchbare Ressource benötigt in jedem Fall zusätzlich zu einer Segmentierung in gesprächslinguistische Grundeinheiten (wie Intonationsphrasen oder Sprechakte) mindestens eine weitere, granularere Segmentierung, im Regelfall in Worteinheiten. Diese Segmentierung findet idealerweise mit der Erstellung einer Normalisierungsebene statt (s. Kap. 2.4.10).

Bitte beachten Sie Arbeitsaufgabe 1 am Ende von Kapitel 2.4.7 für die Erstellung eines Minimaltranskripts bzw. einer literarischen Umschrift in EXMARaLDA nach GAT-Konventionen mit einer Segmentierung in Intonationsphrasen.

2.4.6 | Alignierung von Transkriptionssegmenten mit dem Audiosignal

Die Programme EXMARaLDA, FOLKER, ELAN, PRAAT (und andere Programme) besitzen neben einer Transkriptionsfunktion eine Funktion zum Import von Audiodateien. Beim Transkriptionsprozess können nun Bereiche aus der importierten Audiodatei mit dem transkribierten Text verknüpft werden; man spricht von einer Alignierung des Audiosignals mit der Transkription. Diese Alignierung kann im Prinzip beliebig feinkörnig sein, ist jedoch für das Szenario der Erstellung von Gesprächskorpora ab (gesprächs)linguistischen Grundeinheiten wie Intonationsphrasen, Äußerungen oder Sätzen sinnvoll. Das bedeutet, dass jedes Segment, das beim Transkriptionsprozess als Grundeinheit gilt, mit dem Audiosignal aligniert wird.

Die Alignierung wird bei den verschiedenen genannten Programmen unterschiedlich realisiert, auch weil der Funktionalität der Programme unterschiedliche Nutzerszenarien zugrunde liegen.

Bitte beachten Sie Arbeitsaufgabe 2 am Ende von Kapitel 2.4.7 für die Alignierung des erstellten Minimaltranskripts mit der entsprechenden Audiodatei in EXMARaLDA und FOLKER.

2.4.7 | Weitere lautliche Merkmale im Basistranskript

Neben den bislang behandelten Merkmalen des Minimaltranskripts werden häufig die folgenden artikulatorischen und prosodischen Merkmale annotiert. Zu jedem Merkmal gehört eine eigene Klassifikation, z. B. unterschiedliche Abstufungen von Längen bei der Annotation von Pausen. Diese können jeweils mehr oder weniger stark differenziert werden. Im Basistranskript wird die Differenzierung zugunsten der Handhabbarkeit bewusst gering gehalten.

Akzentstrukturen: Akzente (im Wesentlichen Haupt- und Nebenbetonungen) können einzelnen Wörtern (unabhängig vom Kontext) oder den gesprächslinguistischen Minimaleinheiten (z. B. Intonationsphrasen nach Selting et al. 2009) zugewiesen werden. Im Basistranskript werden nur sogenannte Fokusakzente annotiert (vgl. Selting et al. 2009, S. 19 f.). Hierbei handelt es sich um Silben, die aufgrund einer semantischen Fokussierung hervorgehoben (besonders betont) sind. Diese Hervorhebungen werden mit Großbuchstaben gekennzeichnet. Weil die Großschreibung somit zur Markierung von Fokus dient, kann sie anderweitig nicht verwendet werden. Besonders starke Akzente werden zusätzlich mit Anführungszeichen ›gerahmt‹.

Bitte beachten Sie Arbeitsaufgabe 3 am Ende dieses Kapitels für die Markierung fokussierter Wörter nach GAT 2.

Pausen ~~Pausen~~ innerhalb und zwischen Intonationsphrasen werden gemäß GAT 2 (vgl. Selting et al. 2009, S. 21 f.) bis zu einer Länge von einer Sekunde vierstufig unterteilt: Mikropause bis 0,2 Sekunden (Symbol: .), kurze Pause bis 0,5 Sekunden (Symbol: -), mittlere Pause bis 0,8 Sekunden (Symbol: --), längere Pause bis eine Sekunde (Symbol: ---). Für längere Pausen werden konkrete Werte (z. B. 2.5 für eine zweieinhalbsekündige Pause) angegeben.

Beachten Sie, dass Pausen, die zumeist als Unterbrechung des Sprechflusses definiert werden, ungefüllt (wie in der Interpretation des GAT-Basistranskripts) oder aber auch gefüllt sein können, wobei verschiedene Interjektionen (*ähm*, *äh* usw.) als auch Atemgeräusche und andere Geräusche als sogenannte Filler dienen. Ein Korpus mit der Annotation gefüllter Pausen ist das BeMaTaC-Korpus der Humboldt-Universität zu Berlin (https://hu.berlin/bematac/), in welchem gefüllte Pausen mit dem Annotationswert »f1« auf einer Annotationsebene »df« annotiert wurden, wobei die 1 für die erste Pause innerhalb einer Diskurseinheit steht (Internet-Link zur Suchanfrage im ANNIS-Suchinterface: https://hu.berlin/annis_bematac_filler).

Bitte beachten Sie Arbeitsaufgabe 4 am Ende dieses Kapitels für die Markierung ungefüllter Pausen nach GAT 2.

Tonhöhenverläufe (Prosodie): Tonhöhenverläufe können sich wie Akzente auf den Satz- oder Wortkontext beziehen. Im Basistranskript wird nur die Tonhöhenbewegen am Ende von Intonationsphrasen berücksichtigt (vgl. Selting et al. 2009, S. 21 f.), wobei zwischen fünf Tonhöhenverläufen, gemessen am vorangegangenen Kontext, unterschieden wird: hoch steigend (Symbol: ?), steigend (Symbol: ›), gleichbleibend (Symbol: -), fallend (Symbol: ;) sowie tief fallend (Symbol: .).

Bitte beachten Sie Arbeitsaufgabe 5 am Ende dieses Kapitels für die Markierung verschiedener Tonhöhenverläufe am Ende von Intonationsphrasen nach GAT 2.

Artikulationslänge: Ähnlich wie bei dem Fokus werden ungewöhnlich lang artikulierte Vokale und Konsonanten bei der Transkription gekennzeichnet (vgl. Selting et al. 2009, S. 24). Hierzu dient gemäß den GAT-2-Richtlinien der Doppelpunkt, wobei ein einfacher Doppelpunkt eine Längung bis 0,5 Sekunden, ein zweifacher Doppelpunkt eine Längung von 0,5 bis 0,8 Sekunden und ein dreifacher eine Längung über 0,8 Sekunden markiert.

Bitte beachten Sie Arbeitsaufgabe 6 am Ende dieses Kapitels für die Markierung gedehnter Laute im Wort nach GAT 2.

Schnelle Anschlüsse: Wenn mehrere Intonationsphrasen ohne Pause direkt miteinander verbunden werden, nennt man dies einen schnellen Anschluss. Schnelle Anschlüsse werden mittels Gleichheitszeichen, und zwar jeweils am Ende der vorangegangenen und zum Beginn der folgenden Intonationsphrase (jeweils ohne Leerzeichen zu den umliegenden Zeichen), markiert.

Arbeitsaufgaben

1. Unter der Webadresse https://bit.ly/2OltHuo erhalten Sie eine Audiodatei, die einen dreizehn Sekunden langen Ausschnitt eines Dialogs zwischen zwei Sprecherinnen sowie ein dieser Datei zugeordnetes EXMARaLDA-Transkript enthält. Das Gespräch wird in voller Länge auf der Webseite http://agd.ids-mannheim.de/gat.shtml angeboten; eine Transkription dieses Gesprächs im Sinne des GAT-Basis- und auch des Feintranskripts finden Sie in Selting et al. (2009), S. 42 f. ebenso auf der Webseite. Für die Bearbeitung der folgenden Aufgaben können Sie sich an dem Beispiel des Basistranskripts orientieren; der Übungseffekt ist jedoch stärker, wenn Sie die Umsetzung der Aufgaben zunächst ohne Vorlage versuchen.
   - Entpacken Sie die zwei Dateien in denselben Ordner.
   - Öffnen Sie die EXMARaLDA-Datei.
   - Spielen Sie die Audiodatei in einem separaten Audioplayer vollständig und wiederholt ab.
   - Beginnen Sie mit dem Anfang des Gesprächs: Schreiben Sie die ca. ersten fünf Sekunden des Gesprochenen in literarischer Umschrift in die erste Zelle von Sprecherin 1 in den EXMARaLDA-Partitureditor. Transkribieren Sie bis zu dem Wort *lassen* (erstes Vorkommen). Achten Sie auf kontinuierliche Kleinschreibung und die Kennzeichnung bestimmter Aussprachevarianten durch die Abweichung von der Schreibnorm.
   - Übernehmen Sie das von Sprecherin 2 Gesprochene in die erste Zelle. Sie können die Überlappungen durch die in GAT 2 vorgesehenen eckigen Klammern (auf beiden Ebenen) kennzeichnen oder zunächst den spezifischen Überlappungsbereich unmarkiert lassen.

- Gehen Sie gemäß dem bisherigen Vorgehen bis zum Ende der Audiodatei vor. Die Entertaste erzeugt ein neues Event rechts von dem aktivierten Event. Die Tabstopptaste springt zum nächsten Event, sofern es bereits vorhanden ist.

2. Laden Sie unter der Webadresse https://bit.ly/2FndGQr eine EXMARaLDA- und eine FOLKER-Datei herunter, die mit der Audiodatei im ersten Download oben verknüpft sind. Im folgenden Bearbeitungsschritt geht es um die Alignierung des transkribierten Texts mit den passenden Ausschnitten aus der Audiodatei.
Da EXMARaLDA und FOLKER so konzipiert sind, dass Audiosequenzen beim Transkribieren zugewiesen werden, müssen Sie den in Aufgabe 1 transkribierten Text noch einmal eingeben, während Sie Abschnitte in der Audiodatei zuweisen. Sie können aber die bereits transkribierte Datei separat öffnen und die Zelleninhalte in die neue Bearbeitung hinüberkopieren.
Gehen Sie bei der Bearbeitung folgendermaßen vor:

a) Bearbeitung in EXMARaLDA

- Öffnen Sie die EXMARaLDA-Datei (»Transkript_v1_Audio.exb«).
- Klicken Sie im Editor den Knopf »Append interval«. Sie hören die ersten zwei Sekunden der Aufnahme. Schreiben Sie den entsprechenden Text in die erste Zelle von Sprecherin 1. (Der ausgewählte Ausschnitt entspricht genau der ersten Intonationsphrase im GAT-2-Beispieltranskript. Wenn Sie die erste Einheit verlängern wollen, können Sie dies durch eine Vergrößerung des Audioausschnitts tun.)
- Klicken Sie wieder »Append interval«. Sie hören die nächsten zwei Sekunden der Aufnahme und müssen den Audioausschnitt verlängern (bis ungefähr 4,7 Sekunden in der Aufnahme). Geben Sie den entsprechenden Text ein.
- Fügen Sie bei Sprecherin 2 die Redeanteile hinzu. Achten Sie darauf, dass die ersten zwei Interjektionen (»ja« und »hm«) mit einer Intonationsphrase von Sprecherin 1 überlappen (Sie können diese in zwei Events bzw. Zellen unterteilen) und dass die dritte Interjektion (»hm«) nicht überlappt.
- Führen Sie die Prozedur fort, bis die Aufnahme vollständig bearbeitet ist.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v1_Minimaltranskript.exb«.

b) Bearbeitung in FOLKER

- Öffnen Sie die Datei »Transkript_v1_Audio.flk« mit FOLKER (FOLKER gehört zu den Programmen, die man mit EXMARaLDA auf der Webseite http://www.exmaralda.org herunterladen kann).
- Der erste Transkriptionsausschnitt ist bereits eingerichtet: Klicken Sie auf eine Spalte der ersten angelegten Zeile. Sie können nun den ersten Audioausschnitt abspielen. Geben Sie den entsprechenden Text in die Spalte »Transkriptionstext« ein.
- Klicken Sie ganz rechts auf »Append new segment« (blaues Plus-Symbol mit grünem Pfeil). Hierdurch fügen Sie der Transkription einen neuen Audioausschnitt mit einem neuen Transkriptionsseg-

ment hinzu. Hören Sie den hinzugefügten Ausschnitt, erweitern Sie ihn bis zur nächsten Segmentgrenze und geben Sie den Transkriptionstext in die entsprechende Zeile und Spalte ein.
- Klicken Sie ganz rechts auf »New segment« (blaues Plus-Symbol). Hierdurch fügen Sie demselben Ausschnitt eine neue Transkriptionszeile hinzu. Wählen Sie in der Spalte »Speaker« »Sprecherin 2« und fügen Sie der Zeile das von Sprecherin 2 Gesprochene hinzu (übernehmen Sie beide Interjektionen (»ja« und »hm« in dieselbe Transkriptionszeile).
- Fahren Sie mit der Prozedur fort, bis die Aufnahme vollständig bearbeitet ist. Achten Sie auf die korrekte Zuweisung der Sprecherinnen zu den jeweiligen Beiträgen.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v1_Minimaltranskript.flk«.
- Sie können die gespeicherte Datei mit der Funktion »Import« in EXMARaLDA öffnen und dort weiterbearbeiten (FOLKER ist für das Transkribieren konzipiert, EXMARaLDA für das Transkribieren und weitere Annotieren). Die in EXMARaLDA importierte FOLKER-Transkription sollte identisch mit der in EXMARaLDA erstellten Transkription sein. Ausnahme: Die überlappenden Interjektionen von Sprecherin 2 können in der ursprünglichen EXMARaLDA-Transkription in einzelnen Zellen stehen (Sie können in der von FOLKER nach EXMARaLDA konvertierten Datei die Zelle aufsplitten).
- In den folgenden Aufgaben werden dem bislang erstellten Minimaltranskript weitere bereits besprochene sprachliche Merkmale hinzugefügt und es werden hierfür jeweils eigene Annotationsebenen angelegt.

3. Bauen Sie auf der erstellten Fassung des Minimaltranskripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2JwEcfp). Fügen Sie die Analyse von Akzentsilben hinzu, indem Sie diese mit Majuskeln (Großbuchstaben) markieren. Gehen Sie dabei wie folgt vor:
- Fügen Sie pro Sprecherin je eine Annotationsspur »Akzent« hinzu (bei markierter Annotationszeile STRG-i für »Insert tier« eingeben, Sprecherin zuweisen, Typ »Annotation« wählen, Namen der Spur eingeben, die Inhalte der entsprechenden Transkriptionsspur in die neue Spur kopieren). Ordnen Sie die Annotationsspuren gemäß Abb. 2.20 an.
- Markieren Sie in der hinzugefügten Spur die Akzentsilben mit Majuskeln. Achten Sie dabei auf die Markierung genau der Silbeneinheit.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v2_Akzent.exb«.

Sprecherin_1 [Umschrift]	ja die vierziger generation	so das wahnsinnig viele
Sprecherin_1 [Akzent]	ja die vierziger generation	so das WAHNsinnig viel
Sprecherin_2 [Umschrift]		ja
Sprecherin_2 [Akzent]		ja

Abb. 2.20: Anordnung der Annotationsspuren in EXMARaLDA

4. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2CqbW8n). Analysieren Sie zusätzlich Pausen. Gehen Sie dabei analog zur Prozedur in Aufgabe 3 vor: Fügen Sie den Spuren »Umschrift« und »Akzent« pro Sprecherin jeweils eine Spur »Pausen« hinzu (achten Sie auf die korrekten Einstellungen für die Sprecherzuordnung und den Informationstyp).
   - Entnehmen Sie die Pausentypen dem Dokument Selting et al. (2009), S. 21 f. bzw. den Informationen oben im Kapitel.
   - Fügen Sie die Pausenmarkierungen auf derjenigen Spur derjenigen Sprecherin hinzu, die gerade den Turn besitzt. Trennen Sie die Pausenmarkierungen mit Leerzeichen von Text ab.
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v3_Pausen.exb«.

5. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2JsFSXl). Fügen Sie die Analyse von Tonhöhenverläufen am Ende der Transkriptionssegmente hinzu, indem Sie eine Spur namens »Tonhöhe« hinzufügen und diese mit den Kategorien aus dem Dokument Selting et al. (2009), S. 21 f. versehen.
   - Schreiben Sie die Zeichen ohne Leerzeichenabstand an das letzte Wort einer jeden Zelle. Berücksichtigen Sie also jede als abgeschlossene Intonationsphrase interpretierte Einheit mit einem Annotationswert am Ende der Phrase.
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v4_Tonhoehe.exb«.

6. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2Wgc69U). Fügen Sie eine Analyse der Artikulationslänge von Vokalen und Konsonanten hinzu, indem Sie eine Spur namens »Längung« hinzufügen und diese mit den Kategorien aus dem Dokument Selting et al. (2009), S. 21 f. versehen.
   - Hören Sie durch die Audiodatei und überlegen Sie, welche Laute eine relative Länge gegenüber ihrer Normallautung besitzen. Sie werden feststellen, dass dies relativ schwer zu bemessen ist.
   - Lösungshinweis: Markieren Sie ganz zu Beginn der Aufnahme einen Vokal und in der zweiten Intonationsphrase (nach Selting et al. 2009) einen Konsonanten und einen Vokal, jeweils mit einer einfachen Längung (einfacher Doppelpunkt).
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v5_Laengungen.exb«.

7. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2UNU0vt). Fügen Sie der Gesamtanalyse schnelle Anschlüsse hinzu, indem Sie eine Spur namens »Anschlüsse« hinzufügen und schnelle Anschlüsse von Intonationsphrasen bei derselben Sprecherin analysieren:
   - Markieren Sie jeweils am Ende der vorangehenden Zelle und am Beginn der nachfolgenden mit einem Gleichheitszeichen (ohne

Leerzeichen zu den umliegenden Zeichen) den unmittelbaren Anschluss.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v6.exb«.

2.4.8 | Segmentierung und redundante Information

Die im Aufgabenteil des vorangegangenen Kapitels 2.4.7 zu erstellende komplexe Transkription auf hierarchisch getrennten Beschreibungsebenen enthält in ihrer Gesamtheit zwölf Bearbeitungsebenen, sechs pro Sprecherin (s. Abb. 2.21).

Die entsprechende EXMARaLDA-Transkription können Sie unter der Webadresse https://bit.ly/2WkY7jd herunterladen.

Fügt man sämtliche auf den verschiedenen Ebenen getrennte Informationen zusammen, so erhält man zwei Zeilen, deren Inhalt bis auf Zeilenumbrüche und wenige weitere Merkmale einem GAT-Basistranskript gleicht. Korpuslinguistisch ist die Trennung verschiedener Beschreibungsebenen deshalb ideal, weil die Auswertung der Daten dann viel systematischer erfolgen kann: Man findet spezifische Phänomene auf spezifischen Beschreibungsebenen.

Um die auf den einzelnen Ebenen beschriebenen Phänomene wirklich systematisch finden zu können, ist das im Aufgabenteil von Kapitel 2.4.7 erstellte Transkript noch relativ ungenau. Ziel sollte bei jeder Annotationsvariable sein, pro annotiertem Grundsegment genau einen Fall des annotierten Phänomens zu finden (alle gefüllten Zellen auf der Ebene »Umschrift« sollen genau ein elementares GAT-Segment darstellen, alle gefüllten Zellen auf der Ebene »Pausen« sollen genau eine Pause darstellen usw.). Mit anderen Worten: Auf jeder Beschreibungsebene befindet sich viel überflüssige Information, was vor allem daran liegt, dass beim Erstellen der einzelnen Ebenen zunächst die jeweilige Ebene der Minimaltranskription kopiert wurde, um wiederum das zu annotierende Phänomen möglichst gut lokalisieren zu können. Mit etwas Annotationsübung lässt sich das vermeiden. Für jede Einzelspur ist zu fragen, welche bereits vorhandene Information ggf. dupliziert werden soll.

Abb. 2.21: Screenshot der Annotationsebenen in EXMARaLDA zur Transkriptionsaufgabe in Kap. 2.4.7

Sprecherin_1 [Umschrift]	ja die vierziger generation	so das wahnsinnig viele die sich da ham scheiden lassen	oder scheiden lassen überhaupt		heute noch
Sprecherin_1 [Akzent]	ja die vierziger generation	so das WAHNsinnig viele die sich da ham SCHEIden lassen	oder scheiden lassen Überhaupt		heute noch
Sprecherin_1 [Pausen]	ja ( ) die vierziger generation	so das wahnsinnig viele die sich da ham scheiden lassen	oder scheiden lassen überhaupt		heute noch (2 0)
Sprecherin_1 [Tonhöhe]	ja die vierziger generation-	so das wahnsinnig viele die sich da ham scheiden lassen	oder scheiden lassen überhaupt		heute noch-
Sprecherin_1 [Längung]	ja die vierziger generation	so das: wahnsinnig viele die sich da ham scheiden lassen	oder scheiden lassen überhaupt		heute noch
Sprecherin_1 [Anschluss]	ja die vierziger generation=	=so das wahnsinnig viele die sich da ham scheiden lassen=	=oder scheiden lassen überhaupt		heute noch
Sprecherin_2 [Umschrift]		ja	hm	hm	
Sprecherin_2 [Akzent]		ja	hm	hm	
Sprecherin_2 [Pausen]		ja	hm	hm (--)	
Sprecherin_2 [Tonhöhe]		ja,	hm,	hm,	
Sprecherin_2 [Längung]		ja	hm	hm	
Sprecherin_2 [Anschluss]		ja	hm	hm	

Arbeitsaufgabe

Löschen Sie aus jeder Zelle der EXMARaLDA-Datei »Transkript_v6.exb« (https://bit.ly/2WkY7jd) sämtliche für das jeweilige Phänomen irrelevante Informationen:

- Behalten Sie auf der Ebene »Akzent« nur die Wörter mit Fokusakzent.
- Behalten Sie auf der Ebene »Pausen« nur die Pauseninformationen.
- Behalten Sie auf der Ebene »Tonhöhe« nur die Zeichen für die Intonationskontur am Intonationsphrasenende.
- Behalten Sie auf der Ebene »Längung« nur die Wörter mit Längungen.
- Behalten Sie auf der Ebene »Anschluss« nur die GAT-Segmente mit Anschluss zu einem Nachbarsegment (der Text kann hier beibehalten werden).
- Sie können durch eine geschickte Segmentierung des Gesamttranskripts versuchen, die einzelnen Phänomene möglichst gut mit ihrer Position auf der Ebene des Minimaltranskripts (»Umschrift«) zu verknüpfen. Wortweise lässt sich dies jedoch erst nach einer Tokenisierung der Daten gemäß einer wortbezogenen Tokendefinition erreichen (s. Kap. 2.4.10).
- Speichern Sie die bearbeitete Datei unter dem Namen »Transkript_v7.exb«.

2.4.9 | Nicht-phonetische Merkmale in der Transkription

Neben phonetisch-phonologischen Merkmalen wie den bisher genannten (und vielen weiteren) können auch nicht-sprachliche Merkmale annotiert werden. In der GAT-2-Konverntion sind doppelte runde Klammern für die Kommentierung außersprachlicher Merkmale wie Lachen, Atmen usw. vorgesehen (z. B. ~~auch~~ »((lacht))« usw., vgl. Selting et al. 2009, S. 39 f.). Aus korpuslinguistischer Sicht ist anzumerken, dass spätestens beim Abschluss der Korpuserstellung sämtliche Werte, die auf einer bestimmten Beschreibungsebene vergeben wurden, für die Korpusdokumentation zusammengetragen und den Korpusnutzern gewissermaßen als Tagset zur Verfügung gestellt werden sollten.

Die folgenden häufig auftretenden Merkmale seien exemplarisch erwähnt:

Abbrüche und (Selbst-)Korrekturen: Laut den GAT-2-Richtlinien werden Wortabbrüche mit Glottalverschluss mittels des IPA-Zeichens für den Glottisschlag (ʔ) gekennzeichnet. Wortabbrüche können natürlich allgemein annotiert und die nachfolgenden Wörter als Korrektur innerhalb einer Reparatur gekennzeichnet werden. Belz (2014) beinhaltet Annotationsrichtlinien für Reparaturen mit diversen Beispielen.

Außersprachliche Geräusche: Nicht-sprachliche Geräusche können z. B. Husten, Räuspern oder auch Umgebungsgeräusche usw. sein. Für die Nutzung der Korpusdaten ist es hilfreich, wenn die Annotation solcher Merkmale nicht mit der Analyse der sprachlichen Elemente ver-

mischt wird, sondern gemäß dem Vorgehen im vorangegangenen Kapitel 2.4.7 (Aufgabenteil) auf einer getrennten Beschreibungsebene erfolgt.

Weitere Annotatorenkommentare: In den GAT-2-Richtlinien ist vorgesehen, dass Kommentare zum Sprechmodus in doppelte Spitzklammern geschrieben werden, wobei die innere Klammer den Kommentar und die äußere Klammer den Skopus des Kommentars enthält (z. B. »< <emört> … >«, vgl. Selting et al. 2009, S. 24). Auch hier sollte beim Aufbau eines Korpus die Information auf einer gesonderten Annotationsspur vermerkt werden, damit die Informationen nicht miteinander konfligieren.

2.4.10 | Normalisieren ~~und Tokenisieren~~

Definition

Normalisieren bezeichnet einen Annotationsvorgang, bei dem variierende Formen mit derselben Bedeutung vereinheitlicht werden, um sie später einheitlich finden zu können. Die Normalisierung im Kontext transkribierter Sprachdaten meint die Anpassung dieser Transkriptionsdaten an die Standardorthographie.

Bei einer Transkription, die nicht nach den Regeln der standarddeutschen Orthographie erfolgt, ist eine anschließende Normalisierung für die korpuslinguistische Weiterverarbeitung sowie spätere Korpusnutzung unabdingbar. Dies kann man sich mit einem Blick in die Daten des FOLK-Korpus im DGD-Suchportal (https://dgd.ids-mannheim.de/DGD2Web/) vergegenwärtigen: Stellen Sie sich vor, Sie möchten alle Vorkommen des Verbs *haben* in der 2. Person Singular finden. Wenn Sie in der Transkription suchen, finden Sie für die Form *hast* 2972 Treffer, für *haste* 218, für *hascht* 59, für *hastu* 2. An der Auswahl der Formen sehen Sie: Wir haben kaum eine Möglichkeit, die Gesamtzahl an Vorkommen der abstrahierten Formen *hast* zu ermitteln, wenn wir keine Suchebene zur Verfügung haben, auf der wir mittels eines orthographisch standardisierten Suchausdrucks suchen können. Vernünftigerweise stellt das Korpus eine solche zur Verfügung und wir finden bei der Suche nach der standardisierten Form *hast* 3653 Instanzen (Vorkommen) im Korpus. Gleichzeitig sehen wir, dass unter den ersten Ergebnissen die transkribierten Formen *hoscht* und *hosch* auftreten (das FOLK-Korpus enthält viele Sprecherinnen und Sprecher aus dem Mannheimer Sprachraum), die in der Wortformliste der antizipierten Vorkommen von oben gar nicht enthalten sind.

Anwendungsbeispiel: Bleiben wir bei den nicht normalisierten oben stehenden Varianten von ~~*hast und*~~ *haben* und wenden auf diese Formen (*hascht* usw.) die Standardvariante des TreeTaggers (~~http://~~www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/) an, so kommen wir zu folgenden Taggingergebnissen: *Hast* in dem Satzkontext *Du hast das Buch gefunden.* wird korrekt als finites Hilfsverb (STTS-Tag: VAFIN) analysiert.

Bei den nicht normalisierten Formen *hasch*, *hoscht* usw. bekommen wir hingegen die Tags VVIMP, VVFIN und XY ausgeben, d. h., der Tagger liefert ›falsche‹ Ergebnisse.

Diese Beispiele zeigen die zwingende Notwendigkeit einer Normalisierung: Erstens bietet sie für die Anwendung automatischer Analyseverfahren eine angemessene Datengrundlage, zweitens gewährleistet sie bei der späteren Korpussuche, dass alle Vorkommen von bestimmten Lexemen systematisch gefunden werden können.

Dass eine Normalisierung generell und nicht nur bei der Aufbereitung mündlicher Sprache notwendig ist, leuchtet ein, wenn wir uns vergegenwärtigen, dass auch schriftliche Texte fehlerbehaftet sein können und nicht-standardisierte Formen enthalten können (s. Kap. 2.2.4). Bei der Verarbeitung mündlicher Sprache wird diese Notwendigkeit nur noch sichtbarer, denn hier ist die Normalisierung das wesentliche Bindeglied zwischen der schriftlichen Repräsentation des zugrunde liegenden Sprachsignals (also der Transkription) und der weiteren Verarbeitung dieser Daten.

Annotationswerkzeuge: Zum Normalisieren von sowohl original geschriebenen als auch transkribierten gesprochenen Daten existieren kaum Programme, die sich ad hoc bedienen lassen. Ein Grund dafür ist, dass die Normalisierungsprogramme in erster Linie mithilfe von Normalisierungsdatenbanken arbeiten, die häufig riesig groß oder nicht frei verfügbar sind. Das Programm alleine genügt also oftmals nicht, um Ergebnisse zu erzielen. Außerdem werden von den existierenden Programmen lediglich Vorschläge zur Änderung bestimmter ›auffälliger‹ Wortformen gemacht, was einige manuelle Arbeit nach sich zieht. Diese kann allerdings gut investiert sein, denn Normalisierungsfehler führen häufig zu entscheidenden Folgefehlern. Ein zukunftsweisendes Normalisierungstool ist Thomas Schmidts »OrthoNormal« (http://exmaralda.org/de/orthonormal-de/), das zusammen mit dem EXMARaLDA-Partitur-Editor und dem Transkriptionswerkzeug »FOLKER« erhältlich ist (http://www.exmaralda.org, Schmidt 2012). Am unkompliziertesten erhält man jedoch Normalisierungsvorschläge zu einem beliebigen Nichtstandardtext, wenn man Brian Jurishs Normalisierungsprogramm CAB (Jurish 2012; http://www.deutschestextarchiv.de/demo/cab/) in dem online verfügbaren Annotationsservice WebLicht (https://weblicht.sfs.uni-tuebingen.de/; s. Kap. 2.3) verwendet.

Normalisierungsebene: Die Normalisierung der bislang nicht normgerecht repräsentierten Korpusdaten muss auf einer gesonderten Annotationsebene im Korpus geschehen. Außerdem sollte die Normalisierungsebene eine bestimmte andere Ebene im Korpus als Grundlage nehmen. Nach der Normalisierung kann die Normalisierungsebene zur Weiterverarbeitung der Korpusdaten mit Standardprogrammen wie Taggern verwendet werden. Außerdem können auf der Normalisierungsebene bestimmte Wortformen zuverlässig gefunden werden, selbst wenn sie nicht standardmäßig artikuliert wurden. Auch kann die Normalisierungsebene im Korpus mit nicht-normalisierten Ebenen genutzt werden, um bestimmte Varianten zuverlässig zu finden. Vergleichen Sie z. B. die Suche nach Vorkommen der normalisierten Wortform *haben* mit gleichzeitiger

phonetischer Repräsentation der artikulierten Form als *ham* im BeMaTaC-Korpus: https://hu.berlin/annis_bematac_norm.

Eine Tokenisierung von Korpusdaten in Gesprächskorpora nach Worteinheiten sollte entweder von Beginn an oder aber spätestens auf der Normalisierungsebene erfolgen.

Arbeitsaufgabe

Arbeiten Sie mit der letzten Version des komplexen Transkripts weiter. Die Lösung zur letzten Arbeitsaufgabe in Kap. 2.4.8 erhalten Sie unter der Webadresse https://bit.ly/2Y84DLP. Führen Sie in diesem Dokument die folgenden Arbeitsschritte zur Normalisierung der Daten durch.

- Fügen Sie für die beiden Sprecherinnen eine Normalisierungsebene mit der Bezeichnung »norm« hinzu, wobei Sie den Inhalt der jeweiligen Ebene »Umschrift« kopieren (s. ggf. die Hinweise im Aufgabenteil von Kap. 2.4.7).
- Passen Sie den Inhalt der Transkription an die deutsche Standardorthographie inklusive Zeichensetzung an.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v8.exb« ab.
- Bearbeiten Sie auch die Arbeitsaufgabe im nachfolgenden Kap. 2.4.11, um die Weiterverarbeitung der normalisierten Ebene zu behandeln.

2.4.11 | Grammatische Annotationen

~~Sobald~~ man über eine normalisierte Ebene der Transkription der eigentlichen Sprecheräußerungen verfügt, kann man sämtliche Analyseschritte durchlaufen, die bereits in den Kapiteln zum Oberkapitel 2.2.6 behandelt wurden. Sobald eine schriftliche Repräsentation und deren Normalisierung erstellt wurde, können also alle dort erwähnten Annotationsebenen mit exakt derselben Methodik auf gesprochene Sprache angewendet werden. Bearbeiten Sie hierzu exemplarisch die folgende Arbeitsaufgabe.

Arbeitsaufgabe

Bei der Fertigstellung dieses Buchs existierte noch keine einfache Möglichkeit, beliebige Spuren in EXMARaLDA oder einem der mit EXMARaLDA kompatiblen Programme zu tokenisieren und zu taggen. Andreas Nolda erstellt derzeit eine Anwendung für die EXMARaLDA-Programmversion ›EXMARaLDA (Dulko)‹ (https://andreas.nolda.org/software.html), mit der verschiedene automatisierte Transformationen, u. a. der Taggingprozess mit dem TreeTagger für beliebige Spuren ermöglicht wer-

den. Um nachzuvollziehen, zu welchen Ergebnissen eine Anreicherung von Korpusdaten wie z. B. der Datei »Transkript_v8.exb« (https://bit.ly/2TmwSCI) führen kann, befolgen Sie die folgenden Schritte:

- Laden Sie die Datei »Transkript _v8.exb« unter dem angegebenen Weblink herunter. Sie enthält eine zur anschließend getaggten Datei passende Aufteilung der EXMARaLDA-Grundsegmente (Timeline-Items).
- Laden Sie eine zweite EXMARaLDA-Datei herunter, in welcher die beiden »norm«-Spuren der Datei »Transkript_v8.exb« mithilfe von EXMARaLDA (Dulko) getaggt wurden: https://bit.ly/2ugYcs6. Verschmelzen Sie die beiden Dateien, indem Sie »Transkript_v8.exb« in EXMARaLDA öffnen und die andere Datei mithilfe der Funktion »Transcription« > »Merge transcriptions...« hinzufügen.
- Ordnen Sie die unten hinzugefügten Annotationsebenen den jeweiligen Sprecherinnen zu, indem Sie sie nach oben verschieben (Funktion »Change tier order...« oben im grafischen Menü) und analog zueinander anordnen.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v9.exb«.

2.5 | Evaluation von Korpusannotationen

Sowohl manuelle wie auch automatisch erstellte Annotationen sollten evaluiert werden, bevor die Korpusdaten veröffentlicht bzw. für Auswertungen genutzt werden. Für manche Werkzeuge wurden Korrektheitsraten veröffentlicht, so z. B. beim TreeTagger: In Schmid (1995) werden Werte für die Akkuratheit von ca. 97 % angegeben. Das bedeutet allerdings keinesfalls, dass es sich hierbei um stabile Werte handelt, sondern je nach Beschaffenheit der zu taggenden Daten wird ein Programm sehr unterschiedliche Werte liefern. Dasselbe gilt für menschliche Annotatorinnen und Annotatoren.

2.5.1 | Evaluationsperspektiven

Die existierenden Verfahren zur Messung der Qualität von Annotationen lassen sich grundlegend in zwei Vergleichsperspektiven einteilen: erstens den Vergleich zweier (oder mehrerer) unabhängig voneinander erstellter Analysen, die jeweils fehleranfällig sein können, zweitens den Vergleich einer potenziell fehleranfälligen Analyse und einer ›korrekten‹ Analyse. Hieraus lassen sich verschiedene Aussagen schlussfolgern: Wenn wir zwei menschliche Annotatoren unabhängige Analysen erstellen lassen und diese im Nachhinein vergleichen, können wir ableiten, wie konsistent die Analyse gemäß einem gegebenen Tagset und gegebenen Analyserichtlinien ist. Das heißt, wir können eine Aussage darüber ableiten, wie leicht oder schwer es Annotatorinnen und Annotatoren fällt, dieselbe Analyse hervorzubringen, und evaluieren damit das Handwerkszeug, mit dem gearbeitet wurde: ein Tagset und Hinweise zur Vergabe der Tags. Ist

die Übereinstimmung schlecht, sollte ggf. das Analyseinventar überarbeitet werden. Ist die Übereinstimmung gut, bestätigt dies das Inventar an Tags und die Vergaberichtlinien. Wir nennen das dargestellte Verfahren Inter-Annotator Agreement (IAA). Wenn wir bei denselben Annotatorinnen oder Annotatoren wiederholt ein IAA messen, können wir den Einarbeitungsprozess der Annotatorinnen und Annotatoren darstellen. Ein IAA kann unter bestimmten Bedingungen auch zwischen automatisch erstellten Analysen sinnvoll sein, z. B. um Hinweise auf von den Programmen häufig fehlerhaft analysierte Kategorien zu erhalten. Wie das IAA durchgeführt wird, erfahren Sie in Kapitel 2.5.2.

Wenn wir über eine Analyse verfügen, von der wir wissen, dass sie korrekt ist (auch wenn diese Aussage schwer zu treffen ist), können wir eine beliebige Analyse derselben Daten nehmen und diese gegen die korrekte Analyse evaluieren. Die korrekte Analyse heißt Goldstandard. Die mit dem Goldstandard verglichene Analyse wird durch einen Übereinstimmungswert direkt evaluiert. Diese Evaluation kann sich auf manuell oder automatisch erstellte Analysen gleichermaßen beziehen. Sie kann auch differenziert erfolgen, indem die verschiedenen zu analysierenden Kategorien (die verschiedenen Tags eines Tagsets) getrennt betrachtet werden. Wie die Evaluation gegen einen Goldstandard durchgeführt wird, erfahren Sie in Kapitel 2.5.3.

2.5.2 | Übereinstimmung von Analysen: das Inter-Annotator-Agreement (IAA)

Wie bereits erwähnt, ist das IAA relevant, um eine Aussage über bestehende Analyserichtlinien und Tagsets zu machen: Stellen Sie sich vor, Sie verfassen ein Kategoriensystem, das Sie in einem Tagset zusammenfassen. Sie formulieren Außerdem Regeln, unter welchen Bedingungen die verschiedenen Tags zu vergeben sind (vergleichen Sie z. B. das in Kapitel 2.2.6.1 behandelte STTS-Tagset mit den in Schiller et al. 1999 formulierten Vergaberichtlinien). Die Stärke der Übereinstimmung zweier oder mehrerer Analysen, die dieses Material verwendet haben, gibt ~~unmittelbar~~ Aufschluss über die Handhabbarkeit des Materials. Erreicht man keine gute Übereinstimmung, wird man darüber nachdenken, wie man die Übereinstimmung verbessern kann, indem man gewisse Kategorien zusammenführt, die Kriterien zur Vergabe der Tags eindeutiger gestaltet usw.

2.5.2.1 | Prozentuale Übereinstimmung

~~Der~~ vergleichsmäßig einfache Weg zur Messung der Übereinstimmung von Analysen ist die Angabe einer prozentualen Übereinstimmung sämtlicher vergebener Kategorien oder bestimmter Kategorien. Hierbei wird angegeben, in wie viel Prozent der Fälle, in denen ein bestimmtes Tag vergeben wurde, die Vergabe in unabhängigen Analysen übereinstimmt. Dies wird über die Relation bzw. den Quotienten der übereinstimmenden Fälle zu der Gesamtheit der analysierten Fälle.

Fallbeispiel: Nehmen wir an, es soll die Übereinstimmung beim Wortartentagging durch zwei unabhängige Analysen ermittelt werden. In diesem Fallbeispiel ist die Anzahl der Token die Gesamtzahl der Fälle, denn jedem Token soll ein Wert für eine Wortart zugewiesen werden. Folgende fiktive Werte seien für das Fallbeispiel angenommen.

Tokenzahl	Anzahl Übereinstimmungen	Anzahl Abweichungen
2988	2477	511

In dem fiktiven Beispiel ist das Korpus insgesamt 2988 Token groß und in 2477 Fällen haben wir übereinstimmende Werte. Die Zahl der Abweichungen ergibt sich aus der Differenz von Tokenzahl und der Zahl der Übereinstimmungen. Der prozentuale Wert der Übereinstimmungen entspricht 2477 / 2988 ≈ 0,83. Die prozentuale Übereinstimmung beträgt demnach gerundet 83 %.

Interpretation des Übereinstimmungswerts: Wie bereits erwähnt, lässt ein solcher Wert Rückschlüsse darüber zu, wie sehr Analyserichtlinien, bestehend aus einem Tagset und Anweisungen zur Vergabe der Tags, verständlich oder verstanden sind. Zum Beispiel kann man den fiktiven Wert von 83 % der Wortartübereinstimmung so interpretieren, dass die Annotatorinnen oder Annotatoren einen akzeptablen Wert für das IAA noch nicht erlangt haben. Hinsichtlich der verschiedenen Kategorien, die zur Wortartenannotation vergeben werden müssen, ist der gemittelte Wert nicht aussagekräftig. Man wird versuchen zu ermitteln, welche Kategorien den Gesamtwert negativ beeinflussen, indem man die Übereinstimmung getrennt für die verschiedenen Tags ermittelt. Man kann dies tun, indem man für jedes Tag die Übereinstimmung separat berechnet, so dass man zu der folgenden fiktiven Aufstellung von STTS-Kategorien mit dazugehörigen Vergabe- und Übereinstimmungswerten gelangt.

Tag	Anzahl Vergaben	Anzahl Übereinstimmungen	prozentuale Übereinstimmung
ART	167	164	0,99
APPR	122	117	0,96
...	...	...	...
ADV	175	128	0,73
ADJD	66	45	0,68

In dem dargestellten Szenario liegt nahe, dass sich die Annotatorinnen oder Annotatoren intensiver mit den Richtlinien zur Vergabe der unteren STTS-Tags ADV und ADJD befassen und die Richtlinien ggf. verfeinert oder präzisiert werden, so dass sich die Übereinstimmungsraten bei den problematischen Kategorien verbessern.

Prozentuale Übereinstimmungswerte sind leicht zu erheben und relativ intuitiv zu interpretieren. Sie haben jedoch den Nachteil, dass sie die spezifische Anforderung, die durch die Größe des Tagsets und somit durch die Anzahl der Möglichkeiten pro Tagging-Entscheidung gegeben

ist, nicht miteinbeziehen können. Wenn wir z. B. ein Tagset mit drei Kategorien verwenden, ist natürlicherweise eine höhere Übereinstimmung als bei einem Tagset mit 50 Tags zu erwarten. Ebenso wird eine höhere Übereinstimmung vorliegen, wenn bestimmte Tags besonders häufig vergeben werden, als wenn die Wahrscheinlichkeit für eine Tag-Vergabe bei den verschiedenen Tags des Tagsets relativ gleich ist. Das bedeutet, dass wir die Ergebnisse verschiedener IAA-Auswertungen manchmal nicht direkt miteinander vergleichen dürfen. Komplexere statistische Maße wie z. B. das »Cohens Kappa« sind dafür entwickelt worden, eine höhere Vergleichbarkeit zwischen verschiedenen IAA-Messungen zu bieten. Diese Maße werden in Kapitel 2.5.3.1 behandelt.

Arbeitsaufgabe

- Laden Sie die Datei »Uebereinstimmung_prozentual.xlsx« von der Webadresse https://bit.ly/2FuSRTl herunter. Öffnen Sie die Datei in LibreOffice (oder OpenOffice) Calc oder Microsoft Excel und durchlaufen Sie die folgenden Schritte:
- Klicken Sie in die Zelle E2 und doppelklicken Sie dann auf das Symbol rechts unten in der Zelle. Sie bekommen sämtliche Unterschiede zwischen einer automatisch generierten Wortartenanalyse des TreeTaggers und der manuellen Korrektur angezeigt.
- Berechnen Sie anhand der Gesamtzahl der pos-Vergaben und der Übereinstimmungen die prozentuale Korrektheit des TreeTaggers bzw. die Übereinstimmung in Prozent.

2.5.2.2 | Adjudizieren

Definition

Den Prozess des Angleichens unterschiedlicher Analysen zu einer verbesserten Analyse nennt man **adjudizieren**.

Wenn man das IAA bestimmt hat, kann man die Vergleichsperspektive weiter nutzen, um die eigentliche Analyse der Daten zu verbessern. Hierbei müssen sich die Annotatorinnen oder Annotatoren im Fall von Abweichungen auf eine geltende Variante einigen. Häufig ergibt sich diese, indem eine der Personen einen offensichtlichen Fehler entdeckt und korrigiert. Manchmal jedoch können Unterschiede in der Analyse auf unterschiedlich interpretierbare Strukturen zurückgehen, die wiederum auf nicht eindeutigen Richtlinien beruhen können. Das Adjudizieren kann also sowohl zur Verbesserung der Annotationen als auch indirekt zur Verbesserung der Annotationsrichtlinien beitragen.

Technische Realisierung: Es ist sinnvoll, beim Adjudizieren ein Vergleichsprogramm oder eine programminterne Vergleichsfunktion zu nutzen, durch die man systematisch von Unterschied zu Unterschied springen kann. Einige der verfügbaren Programme zur Visualisierung von Unterschieden sind nicht speziell für Korpusdaten, sondern ganz allgemeine Zwecke entwickelt worden. Dies gilt z. B. für die Programme Meld (http://meldmerge.org/) und KDiff3 (http://www.fosshub.com/KDiff3.html). Vergleichsprogramme und in Annotationsprogrammen implementierte Vergleichsfunktionen existieren z. B. für Dependenzannotationen, weil es hier nützlich ist, nicht nur Unterschiede in den Quelldaten aufzuzeigen, sondern diese Daten zusätzlich zu visualisieren. What's Wrong With My NLP (https://code.google.com/archive/p/whatswrong/downloads) oder die Vergleichsfunktion von WebAnno (https://webanno.github.io/webanno/downloads/) sind Beispiele für Werkzeuge, die die Unterschiede in mehreren Parses zu denselben Daten anzeigen.

2.5.3 | Evaluation gegen einen Goldstandard

Definition

> Korpusannotationen, die als fehlerfrei (hinsichtlich bestimmter Merkmale) gelten, bezeichnet man als **Goldstandard** (oder auch Ground Truth).

Solche Daten gehen häufig aus mehreren Iterationen (Wiederholungen) von Korrekturdurchläufen und/oder IAA-Durchläufen hervor und werden vor allem zum Trainieren automatischer Annotationsprozesse, z. B. von Taggern und Parsern, benötigt. Sie können aber auch dazu verwendet werden, menschlich und maschinell erstellte Annotationen zu evaluieren.

Umgekehrt formuliert, können wir erst dann etwas über die Güte einer Annotation aussagen, wenn wir sie an einer korrekten Analyse, einem Goldstandard, messen können. Existiert dieser, so bestehen im Grunde dieselben statistischen Möglichkeiten, die Güte der nicht perfekten Analyse in Zahlen auszudrücken, wie sie im Kontext des Inter-Annotator-Agreements kurz angerissen wurden (s. Kap. 2.5.2).

2.5.3.1 | Berechnung von Precision, Recall und F-score

Definition

> Die Häufigkeit, mit der eine vergebene Kategorie korrekt analysiert wurde, nennt man **Precision**. Die Häufigkeit, mit der alle zu einer bestimmten Kategorie gehörigen Instanzen tatsächlich dieser Kategorie zugeordnet wurden, nennt man **Recall**. Die Mittelung aus beiden Größen nennt man **F-score**.

Wie bei der Berechnung des Inter-Annotator-Agreements gezeigt, kann man auch bei der Evaluation von Analysen gegen einen Goldstandard prozentuale Werte ermitteln. Manchmal ist es aber sinnvoll, die Annotationsgüte in verschiedene Aspekte (Gütekriterien) zu zerlegen, um ein differenziertes Bild von der Akkuratheit einer Analyse zu erhalten. Stellen Sie sich vor, Sie möchten evaluieren, wie gut ein Tagger eine bestimmte Wortartkategorie, z. B. Verbpartikeln analysiert, weil Sie vorhaben, diese Wortart mithilfe automatischen Taggings ausfindig zu machen und anschließend zu untersuchen. Die Akkuratheit des Taggers hinsichtlich dieser Kategorie wird durch zwei Fragen bestimmt, die ganz unterschiedlich beantwortet werden können:

1. Wie viele der vom Tagger als Verbpartikeln identifizierten Einheiten sind tatsächlich Verbpartikeln?
2. Wie viele der in den Daten vorhandenen Verbpartikeln hat der Tagger womöglich verpasst?

Werte für die Präzision (Precision) beantworten die Frage 1., Werte für den Recall beantworten die Frage 2. Der F-score ist ein Wert für das abwägende Mittelmaß aus beiden Werten.

Mithilfe eines Goldstandards kann man abgleichen, wie viele Einheiten einer bestimmten Kategorie korrekt annotiert wurden und wie viele verpasst wurden. Nehmen wir die folgende Datenlage für das Verbpartikel-Szenario an.

- In einem gegebenen Korpus kennzeichnet der zu evaluierende Tagger 834 Token als Verbpartikeln.
- Im Abgleich mit zur Verfügung stehenden Goldstandarddaten ergibt sich, dass 806 dieser Token tatsächlich Verbpartikeln sind. Der Tagger hat aber auch 412 Verbpartikeln verpasst und als andere Wortarten ausgewiesen.

Precision: Der prozentuale Wert für die Präzision des Taggers ergibt sich aus dem Quotienten der 806 korrekt analysierten Token und der Gesamtzahl 834 der als Verbpartikeln analysierten Token inklusive der sogenannten ›False Positives‹ (Taggingfehler). Der gerundete Wert ist 0,97. Der Tagger besitzt auf den analysierten Daten also eine Taggingpräzision für Verbpartikeln von gerundet 97 %.

Recall: Der prozentuale Recall-Wert wird ermittelt, indem die 806 tatsächlich gefundenen Verbpartikeln durch die Gesamtzahl der im Korpus vorhandenen Verbpartikeln (das sind 806 + 412 = 1218) geteilt werden. Der resultierende Wert ist 0,66 und die daraus abgeleitete Aussage lautet, dass der Tagger nur ca. 66 % und somit zwei Drittel der im Korpus vorhandenen Verbpartikeln hat ausfindig machen können.

F-score: Die einfachste Formel für die Berechnung des zwischen Precision und Recall vermittelnden F-scores ist

$$F = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}$$

Gemäß den ermittelten Daten ergibt sich somit ein gerundeter F-score-Wert von 0,79, das harmonische Mittel des relativ hohen Precision-Werts und des relativ niedrigen Recall-Werts.

In den meisten Fällen werden auf diese Weise Werte für die Akkuratheit (Accuracy) von Annotationsprogrammen wie Taggern ermittelt und angegeben.

Arbeitsaufgabe

- Laden Sie das Ergebnis der Eigennamenkorrektur (s. Kap. 2.2.6.8; die Datei »Eigennamenerkennung_TreeTagger_korrigiert.exb«) in EXMARaLDA. Sie erhalten die Datei auch unter der Webadresse https://bit.ly/2FnX7UB.
- Ermitteln Sie anhand der Annotationsebenen »pos« (das ist die automatische Analyse des TreeTaggers, mithilfe von TagAnt erzeugt) und »pos_korrigiert« (der manuellen Korrektur) die Werte von
  a) Precision,
  b) Recall,
  c) F-score

  für die Erkennung von Eigennamen durch den TreeTagger.

2.6 | Daten über die Daten: Annotation von Metadaten

Definition

Metadaten bezeichnen sämtliche Informationen über die eigentlichen Korpusdaten.

Sie können sich auf jegliche Aspekte der Beschaffenheit der sogenannten Primärdaten beziehen, die der Korpusaufbereitung zugrunde liegen. So können Metadaten z. B. Aufschluss über Eigenschaften der Textproduzentinnen und Textproduzenten geben (etwa Name, Alter, Geschlecht) oder den im Korpus analysierten Text klassifizieren (etwa Textsorte, Textproduktionssituation, Textmedium usw.). Metadaten können sich aber auch auf den gesamten Prozess der Korpuserstellung beziehen: Annotatoren, Typen von Annotationen, Annotationswerkzeuge und -richtlinien, Tagsets und viele andere Merkmale des Erstellungsprozesses können in den Metadaten spezifiziert werden. Da viele dieser Informationen innerhalb eines Korpus variieren können, müssen sich Metadaten nicht auf sämtliche Daten im Korpus beziehen, sondern können jeweils beliebig große Ausschnitte klassifizieren.

Metadaten vs. Annotationen: Die Offenheit hinsichtlich der Informationen, die als Metadaten gelten sollen, führt dazu, dass im Grunde genommen Auslegungssache ist, was als Metadatum im Korpus gespeichert wird und was als Annotation. Oder anders formuliert: Der Übergang von Metadatum zu Annotation ist fließend. Typischerweise sind Annotationen linguistische Interpretationen, die man klassischen linguistischen Teilgebieten wie der Syntax zuordnen kann, wohingegen Metadaten Fakten über die Daten sind, die diesen Analysen zugrunde liegen. Doch stellen Sie sich vor, Sie unterteilen einen korpusbasiert aufbereiteten Text in Einleitung, Hauptteil und Schluss – je nach Hintergrund bzw. Motivation dieser Analyse kann man diese Tätigkeit als textlinguistische Annotation oder als Metadatum zu den Korpusdaten interpretieren.

Auch die angesprochene Faktizität der Informationen ist relativ: So kann die Zuordnung eines Textes zu einer bestimmten Textsorte wie Kurzgeschichte als anfechtbare oder vage Interpretation angesehen werden. In gewissen Kontexten kann sie auch als Fakt interpretiert werden, wenn z. B. die Autorin oder der Autor den Text selbst als eine bestimmte Textsorte ausgewiesen hat. Linguistinnen und Linguisten müssen sich immer wieder darüber klarwerden, dass jede Kategorisierung eine potenziell streitbare Interpretation ist. Somit kann es keinen klaren Unterschied zwischen Metadaten und Annotation geben, was nahe legt, Metadaten als eine spezielle Art von Annotation aufzufassen. Entsprechend wurde in Kapitel 2.4.1 bereits festgestellt, dass auch die Transkription nichts anderes als eine bestimmte Annotation ist.

Speicherung von Metadaten: Metadaten werden normalerweise dokumentbezogen gespeichert und sind somit in den Korpusdaten klar von den Korpusannotationen getrennt. Wie bei den Annotationen selbst existiert kein einheitliches Format dazu. In XML-kodierten Korpusdaten befinden sich die Metainformationen zumeist in sogenannten Headerdaten, also innerhalb derselben Datei, die auch die Annotationen und den Annotationen zugrunde liegenden Primärdaten enthält. Viele Annotationswerkzeuge erlauben es, zusätzlich zu der Erstellung von Annotationen auch Metadaten einzupflegen. So kann man z. B. in EXMARaLDA über die Funktion »Meta information« beliebige Einträge für Variablen (»Attribute«, z. B. ›Alter‹) und Werten (»Value«, z. B. ›28‹) vornehmen.

Für die Korpussuche unter Berücksichtigung von Metadaten s. Kapitel 3.1.2.27.

3 Praxisteil II: Suchinterfaces, Anfragesprachen und Anfragemöglichkeiten

3.1 Suchwerkzeuge für eigens erstellte Daten
3.2 Online-Suchinterfaces für große Standardkorpora
3.3 Evaluation von Korpussuchen

In den folgenden Kapiteln werden Ihnen verschiedene Möglichkeiten vorgestellt, mithilfe von Suchinterfaces bereits verarbeitete Korpusdaten durchsuchen und Suchergebnisse erhalten zu können. Suchinterfaces sind computer- oder serverbasierte Schnittstellen, mit denen Sie mithilfe einer bestimmten Anfragesyntax auf Korpusdaten zugreifen können und die Ihnen Wege zur Ansicht und zum Export von Treffern bieten. Hierbei werden die vorgestellten Programme grob in zwei Typen unterschieden, die kurz erläutert werden sollen.

Unterschiede zwischen den vorgestellten Suchwerkzeugen: Die in Kapitel 3.1 eingeführten Suchwerkzeuge (AntConc, ANNIS, CQP und NoSketch Engine und TIGERSearch und TüNDRA) und in Kapitel 3.2 behandelten (DWDS und DTA, COSMAS II und KorAP sowie DGD) werden aus den folgenden Gründen getrennt behandelt: Erstgenannte sind als dezentral distribuiert, korpusunspezifisch, annotationsunspezifisch und anfragesprachemächtiger anzusehen, während die übrigen Ressourcen als zentral distribuiert, korpusspezifisch, annotationsspezifisch und weniger anfragesprachenmächtig einzustufen sind. Dies gilt tendenziell und beinhaltet Ausnahmen, die kurz erläutert werden.

Zu AntConc, ANNIS, CQP und NoSketch Engine und TIGERSearch und TüNDRA: AntConc, ANNIS, CQP und NoSketch Engine und TIGERSearch sind frei verfügbare Korpussuchprogramme, die losgelöst von spezifischen Korpora angeboten werden. Die Nutzerinnen und Nutzer können jeweils Korpora mit beliebig vielen Annotationen in die Programme einlesen und mit den angebotenen Such- und Statistikfunktionen bearbeiten. Vorgegeben ist jeweils ein bestimmtes Eingabeformat, in das die zu verarbeitenden Korpusdaten gebracht werden müssen. Da also keine bestimmten Korpora verarbeitet werden, sind die Suchmöglichkeiten möglichst offen gestaltet. Das heißt, es sollen beliebige Annotationsebenen verarbeitet werden und beliebig miteinander verknüpft werden können. Dies wiederum bedingt, dass die Anfragesprachen wie Programmiersprachen funktionieren und gelernt werden müssen.

Ein besonderes Programm in der ersten Gruppe ist AntConc, das fast ausschließlich für die Verarbeitung unannotierter Korpusdaten gedacht ist und dessen Funktionen durch Auswahlmenüs vorgegeben sind. Das Programm ist das am häufigsten verwendete frei verfügbare, lokal installierbare Korpusprogramm überhaupt.

TüNDRA existiert in einer zentralen Instanz, in der viele Baumbanken gesammelt sind. Es wird in der ersten Gruppe von Programmen behan-

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_3

delt, weil es gewissermaßen die Weiterentwicklung von TIGERSearch ist und die Anfragesprachen beider Programme identisch sind.

Zu DWDS und DTA, COSMAS II und KorAP und DGD: KorAP in der zweiten Gruppe ist ein sehr mächtiges Suchprogramm, welches sogar mit unterschiedlichen Anfragesprachen bedient werden kann. Außerdem ist es prinzipiell auch außerhalb einer zentralen Instanz als Suchprogramm herunterladbar und mit beliebigen Korpusressourcen nutzbar. Es wurde dennoch der zweiten Gruppe zugeordnet, weil es als Nachfolgeprodukt von COSMAS II entwickelt wird, dieselben (und zusätzliche) Funktionalitäten besitzen wird und sich noch im Entwicklungsstadium befindet. Für die anderen Vertreter der zweitgenannten Gruppe gilt, dass es sich um Onlineportale handelt, die für die Nutzung bestimmter Korpora erstellt wurden (die also relativ parallel zum Aufbau bestimmter Korpora entwickelt wurden). Diese Ressourcen in lokalen Instanzen oder auf anderen Servern für beliebige Korpusdaten nutzbar zu machen, war nicht vorgesehen. Die Werkzeuge sollen für alle in der jeweiligen Instanz verarbeiteten Korpora möglichst so nutzbar sein, dass mit denselben Suchanfragen über alle Korpusdaten hinweg gesucht werden kann.

Die Vertreter der einzelnen Gruppen komplementieren sich somit ideal, weil die Linguistik sowohl Suchwerkzeuge für große Standardkorpora mit niedrigschwelligen Bedienungsfunktionen benötigt als auch Suchwerkzeuge für spezielle Korpora, die ohne bestimmte Beschränkungen verarbeitet werden sollen.

3.1 | Suchwerkzeuge für eigens erstellte Daten

Definition

> Ein **Suchwerkzeug** ist eine computergestützte Anwendung, die die systematische Suche in bereits aufbereiteten Korpusdaten erlaubt. Genauer unterscheiden kann man zwischen **Suchprogrammen** – eigenständigen lokalen oder serverbasierten Anwendungen – und **Suchinterfaces** – Nutzeroberflächen, die die Kommunikation mit einem dahinterliegenden Programm gewährleisten.

Der erste Anwendungsfall für Korpussuchen ist, dass eigene Korpusdaten aufbereitet wurden und diese nun durchsucht werden sollen. Dies ist fast nie mit demselben Programm möglich, das beim Annotieren der Korpusdaten verwendet wurde. Die Welt der korpuslinguistischen Programme muss also klar unterschieden werden in Editoren auf der einen Seite und Suchwerkzeuge auf der anderen. Zusätzlich existieren für komplexere statistische Auswertungen ebenso gesonderte Programme, womit Auswertungswerkzeuge für Korpora entweder auf die Ermittlung von Suchbelegen und Belegfrequenzen oder auf die statistische Auswertung spezialisiert sind.

Zunächst werden wir in Kapitel 3.1.1 die Suchmöglichkeiten des Programms AntConc für unverarbeitete bis rudimentär verarbeitete Korpus-

daten behandeln. Anschließend wird in Kapitel 3.1.2 beschrieben, wie man mit dem Programm ANNIS in tief annotierten Korpusdaten sämtliche Annotationen berücksichtigen und in komplexen Korpussuchen zueinander in Beziehung setzen kann.

3.1.1 | AntConc als Suchwerkzeug

Das von Laurence Anthony entwickelte Korpuswerkzeug AntConc (http://www.laurenceanthony.net/software/antconc/ Anthony 2018) kann als rudimentäres Suchwerkzeug für verhältnismäßig gering verarbeitete Korpusdaten bzw. einfach unverarbeitete Textdaten genutzt werden.

Hauptfunktionen von AntConc: Mit diesem Programm kann man beliebig viele Textdateien einlesen, über den gesamten Inhalt dieser Dateien Oberflächensuchen formulieren und sich die Treffer in ihrem sprachlichen Kontext anzeigen lassen. Man kann ebenso Treffer (mit ihrem Kontext) exportieren. Abb. 3.1 zeigt die grafische Oberfläche des Programms.

Die KWIC-Ansicht: Die meisten Korpussuchmaschinen verfügen über eine Trefferansicht wie die in Abb. 3.1 gezeigte, eine sogenannte KWIC-Ausgabe (Key Word in Context). Auf diese Weise können die Suchtreffer in ihrer textlichen Umgebung betrachtet werden. Bei der in Abb. 3.1 gezeigten Trefferansicht ist zu erkennen, dass die einzelnen Treffer nicht vollständig dem eingegebenen Suchbegriff entsprechen müssen (in Abb. 3.1 werden groß und kleingeschriebene Instanzen gefunden). Dies lässt sich jedoch in jeder korpuslinguistischen Suchmaschine ändern, so dass ausschließlich Wortformen, die genau dem eingetragenen Suchbegriff entsprechen, gefunden werden. Wann die eine oder die andere Einstellung sinnvoll ist und wie man prinzipiell zwischen exakten Suchen und Suchen mit variablen Treffern unterscheidet, wird ab Kapitel 3.1.2 hinsichtlich verschiedener Suchprogramme behandelt.

Abb. 3.1: Screenshot des Korpuswerkzeugs AntConc mit der Suche nach der Oberflächenform *Schloß* über zwei unverarbeitete Textdateien (Franz Kafkas »Das Schloß« und Kurt Tucholskys »Schloß Gripsholm«).

3.1.2 | Exemplarische Suchen in ANNIS, CQP/NoSketch Engine und TIGERSearch/TüNDRA

In den folgenden Kapiteln werden fünf verschiedene Suchprogramme mit ihrer jeweils eigenen Anfragesyntax vorgestellt. Da in CQP und NoSketch Engine sowie TIGERSearch und TüNDRA die Anfragesprachen nahezu identisch sind, müssen pro Suchszenario nur jeweils drei (synonyme) Suchausdrücke gegenübergestellt werden. Somit können wir eine relevante Menge von offline und online verfügbaren Korpussuchprogrammen abdecken, die eher für den Einsatz variierender Korpusressourcen konstruiert wurden (im Anschluss folgt in Kap. 3.2 die Vorstellung online verfügbarer Standardsuchmaschinen mit relativ stabilem Inhalt). Durch die Gegenüberstellung kann außerdem die generelle Suchlogik besser erfasst werden. Zunächst werden die kontrastierten Suchprogramme kurz beschrieben.

ANNIS (Krause/Zeldes 2016; http://corpus-tools.org/annis/) wurde ursprünglich innerhalb des Sonderforschungsbereichs 632 Informationsstruktur (http://gepris.dfg.de/gepris/projekt/5485900) entwickelt, in dem ein bis dahin nicht verfügbares Suchsystem für die sehr unterschiedlichen aus dem Forschungsverbund hervorgehenden Korpusdaten geschaffen werden sollte. Nach Ablauf des Sonderforschungsbereichs wurde ANNIS stetig weiterentwickelt und dient mittlerweile diversen Korpusressourcen als Suchplattform. ANNIS kann praktisch alle gängigen Annotationsformate verarbeiten und miteinander in Beziehung setzen und verfügt über verschiedene Exportfunktionen sowie eine Statistikfunktion zur variablen Erstellung von Frequenzlisten. Das Suchprogramm besitzt eine eigene Anfragesyntax (genannt AQL, ANNIS Query Language, http://corpus-tools.org/annis/aql.html), die hinsichtlich Dominanzrelationen an die TIGERSearch-Anfragesprache (s. u.) angelehnt ist.

Frei zugängliche Versionen von ANNIS zur Recherche in bestimmten Korpora befinden sich an der Humboldt-Universität zu Berlin (https://hu.berlin/annis; hier hat man Zugriff auf verschiedene historische Korpora, Demo-Korpora und verschiedene Lernerkorpora); gesondert davon existiert eine ANNIS-Instanz mit ausschließlich historischen Korpora des Projekts ›Referenzkorpus Altdeutsch‹ (https://hu.berlin/annis-ddd) sowie eine Instanz mit DaF-Lernerdaten (https://hu.berlin/annis-falko). Eine weitere ANNIS-Instanz liegt hinter dem Webinterface des Projekts MERLIN zur Analyse von Lernersprache (http://www.merlin-platform.eu/). An der Georgetown University befindet sich ebenso eine ANNIS-Instanz (https://corpling.uis.georgetown.edu/annis/) mit einigen deutschsprachigen Korpusressourcen (eine Übersicht: https://corpling.uis.georgetown.edu/annis-corpora/). An der Universität Bochum liegen die im Projekt ›Referenzkorpus Mittelhochdeutsch‹ aufbereiteten Texte in einer eigenen Instanz (http://www.linguistics.rub.de/annis/annis3/REM/) und an der Universität Hamburg die Korpora des ›Referenzkorpus Mittelniederdeutsch‹ (http://annis.corpora.uni-hamburg.de:8080/gui/ren). Weitere Instanzen sind aktuell im Entstehungsprozess.

Unter der Internetadresse https://hu.berlin/annis-intro finden Sie eine Instanz, in welcher eine Auswahl von Korpusdaten durchsuchbar ist, die auf die kommenden Inhalte dieses Buchs zugeschnitten sind.

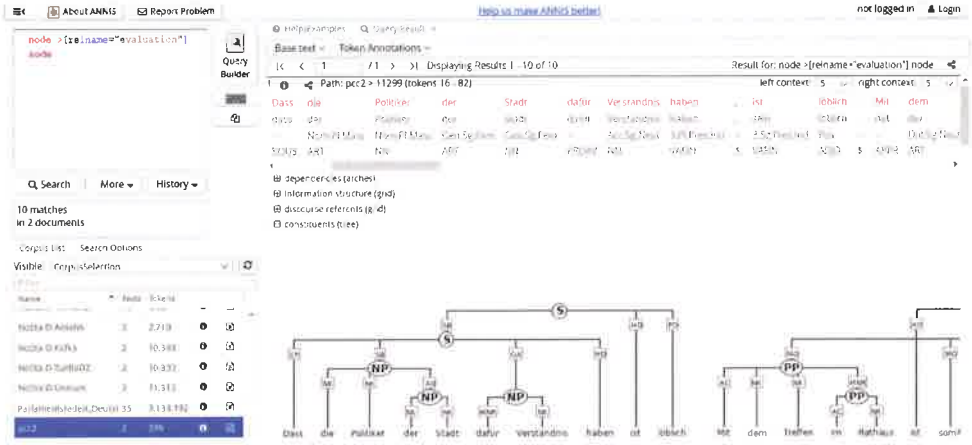


Abb. 3.2: ANNIS-Nutzeroberfläche des Online-Suchinterfaces der Humboldt-Universität zu Berlin (https://hu.berlin/annis)

Abb. 3.2 zeigt die ANNIS-Nutzeroberfläche, in der links oben im Fenster eine Suchanfrage formuliert ist, die in dem links unten ausgewählten Korpus zehn Treffer erzielt (diese Anzahl erscheint nach dem Suchvorgang zwischen dem Sucheingabefenster und der Korpusauswahlliste). Zu jedem Korpus werden die dort enthaltenen Annotationen und Metadaten unter dem Informationsknopf »i« neben Korpusnamen und Korpusgröße in Token bereitgestellt. In dem großen zentralen Fenster werden Suchergebnisse angezeigt, in denen je nach Komplexität der jeweiligen Korpusannotationen gewisse Analysen (wie im Schaubild die dargestellte Syntaxannotation) auf Wunsch sichtbar gemacht werden können. Grundlegende Informationen zum Suchinterface und zur Anfragesprache erhält man unter dem Menü »Help/Examples«. Weitere Funktionen des Interfaces werden an relevanten Stellen zum Datenexport (s. Kap. 3.1.2.28) usw. erwähnt.

Die Suche kann nicht nur per Texteingabe (im Fenster oben links) erfolgen, sondern auch mit einem graphischen Modus im »Query Builder« (das Symbol rechts neben dem Texteingabefenster).

CQP (Evert/Hardie 2011; http://cwb.sourceforge.net/) ist die Kernkomponente einer Reihe von Werkzeugen, die unter dem Namen »IMS Open Corpus Workbench« zusammengefasst werden. Diese Toolbox ermöglicht schnelle Suchen in sehr großen, relativ flach annotierten Korpora. CQP wird auf verschiedenen Universitätsservern betrieben, um große Datenmengen durchsuchen und statistisch auswerten zu können. Hierbei wird das CQP-Suchsystem, das über die Kommandozeile bedient werden kann, häufig durch eine grafische Nutzeroberfläche (*graphical user interface* – GUI) gestützt, um die Bedienung zu erleichtern bzw. die Anwendung allgemein zugänglich zu machen. Hierdurch wird das System zwangsweise auf bestimmte wesentliche Such- und Exportfunktionen limitiert.

Beispiele für CQP-Interfaces (mit verschiedener Mächtigkeit und verschiedenen Korpusdaten) sind das von der Humboldt-Universität betriebene CQP-Interface (https://korpling.german.hu-berlin.de/cqpwi/; Registrierung erforderlich, freie Lizenz für Forschende), das CQP-Interface der TU Dresden (http://linguistik.zih.tu-dresden.de/corpus/; Registrierung er-

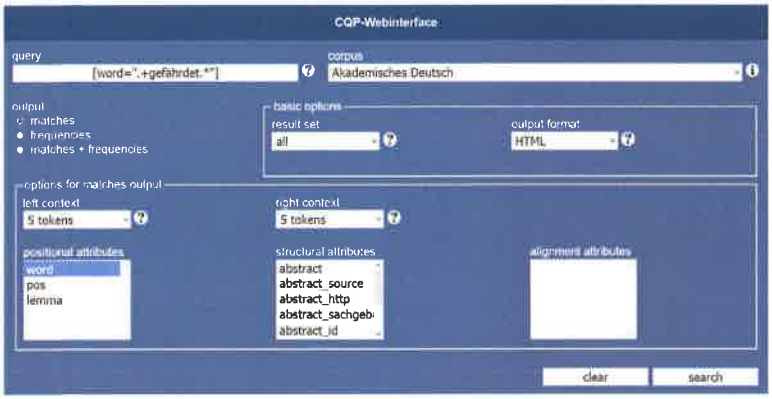


Abb. 3.3: CQP-Nutzeroberfläche des Online-Suchinterfaces der Humboldt-Universität zu Berlin (https://hu.berlin/cqp)

forderlich, freie Lizenz für Forschende), verschiedene über CQP abfragbare Korpora an der Universität Erlangen (https://corpora.linguistik.uni-erlangen.de/demos/CQP/; Demoseite frei zugänglich) sowie diverse Korpora verschiedener Sprachen an der Universität des Saarlandes (https://corpora.clarin-d.uni-saarland.de/cqpweb/; Registrierung erforderlich).

Sie können das Interface zum Testen der unten besprochenen Suchmöglichkeiten mit diesen Zugangsdaten nutzen: username: CQP_Demo, password: TestSuchen. Abb. 3.3 zeigt links oben im Interface eine formulierte Suchanfrage und rechts daneben das für die Suche ausgewählte Korpus. Unter dem Sucheingabefenster kann eingestellt werden, ob Treffer im Textkontext angezeigt werden sollen oder ob Frequenzen zu dem in der Suche erfragten Element ermittelt werden sollen. Für die Ausgabe der Treffer im Kontext (KWIC) kann in der Mitte die linke und rechte Weite des Kontexts eingestellt werden. Links unten werden alle verfügbaren Annotationsebenen im Korpus angezeigt und es kann diejenige ausgewählt werden, die in den Suchergebnissen angezeigt bzw. frequenzmäßig ausgewertet werden soll. Die beiden Fenster unten Mitte und unten rechts führen im Korpus enthaltene weitere Informationen auf (Metainformationen, Bereichsinformationen und relationale Annotationen), die ebenso bei der Suchanfrage und der Darstellung der Ergebnisse berücksichtigt werden können.

NoSketch Engine (https://nlp.fi.muni.cz/trac/noske) ist eine nicht kommerzielle Version des Programms bzw. des Dienstes Sketch Engine (http://www.sketchengine.co.uk/). Eine lokale Installation ist unter bestimmten technischen Voraussetzungen möglich, für Apple- und Windows-Systeme und Nutzerinnen und Nutzer ohne computerlinguistischen Hintergrund jedoch nicht empfehlenswert. Es ist absehbar, dass sich öffentlich zugängliche Korpusinstanzen, die auf der Grundlage von NoSketch Engine zugänglich gemacht werden, zukünftig häufen.

Eine Instanz des Suchprogramms verwaltet die von Felix Bildhauer und Roland Schäfer erstellten COW-Korpora (http://corporafromtheweb.org/). Hier können riesige Webkorpora, u. a. das deutsche DECOW (https://bit.ly/2USzKsQ), online durchsucht werden. Eine vorherige Registrierung ist erforderlich. Studierende können leider nur unter be-

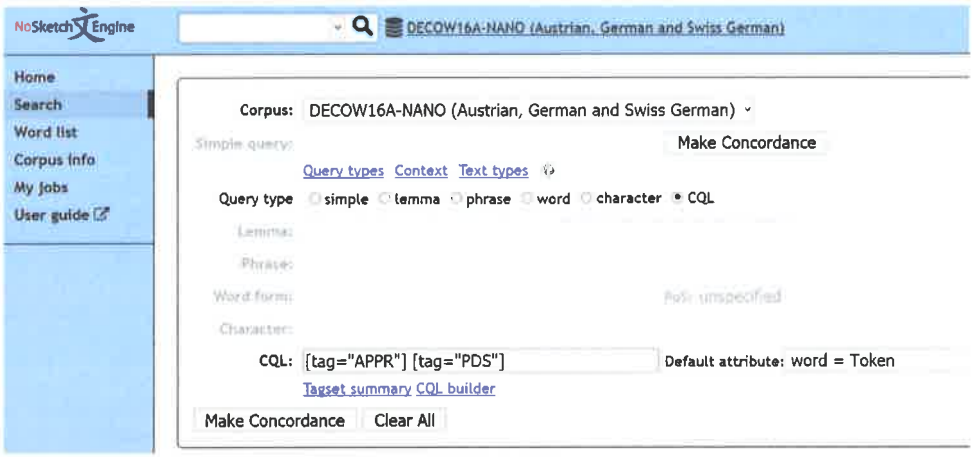


Abb. 3.4: NoSketch-Engine-Nutzeroberfläche des Online-Suchinterfaces der COW-Gruppe (Roland Schäfer und Felix Bildhauer; http://www.webcorpora.org/)

stimmten Bedingungen (https://bit.ly/2FmNZiM) eine Registrierung erlangen.

Abb. 3.4 zeigt das Suchinterface für die COW-Korpora (Schäfer 2016b) nach dem Login der Nutzerin oder des Nutzers. Im hellblauen Bereich erfolgen sämtliche Sucheinstellungen: Oben wird eines der verfügbaren Korpora ausgewählt, darunter wird eine Suchanfrage formuliert, wobei auf verschieden komplexe Suchmöglichkeiten zurückgegriffen werden kann. Die gegebene Suche wird in der Anfragesprache CQL (Corpus Query Language) gestellt, die weitestgehend der CQP-Anfragesprache entspricht, in die weiter unten eingeführt wird.

TIGERSearch (https://bit.ly/2TkhTsV) ist (wie TüNDRA, s. u.) ein Suchwerkzeug für das Durchsuchen von Baumbanken. TIGERSearch wird allerdings nicht mehr gewartet bzw. weiterentwickelt. Es wird lokal installiert und besitzt Such-, Visualisierungs- und Exportfunktionalitäten, die bislang kaum ein Konkurrenzprodukt erreicht.

Um TIGERSearch zu installieren und Korpusdaten hineinzuladen, gehen Sie wie folgt vor.

Anleitung

- Installieren Sie TIGERSearch (https://bit.ly/2TkhTsV) lokal auf Ihrem Computer und importieren Sie mittels des dort mitgelieferten Programms TIGERRegistry das TIGER-Korpus (Version 2012) von der entsprechenden Download-Seite (https://bit.ly/2U1DLxK).
  Noah Bubenhofer stellt auf seiner Webseite sehr hilfreiche Informationen zu den Installationsvorgängen bereit: https://bit.ly/2Omu5c0).
- Alternativ können Sie die Webseite http://fnps.coli.uni-saarland.de:8080/query besuchen. Hier können Sie online auf das TIGER-Korpus zugreifen, allerdings fehlen hier wesentliche Funktionen zum Exportieren der Suchergebnisse und zur statistischen Bearbeitung der Treffer, die das Programm TIGERSearch bietet.
- Auch im ANNIS-Interface der Humboldt-Universität zu Berlin ist das TIGER-Korpus verfügbar: Betätigen Sie den Internetlink https://bit.ly/2TTkYp2. Sie müssen zunächst die Nutzungsbedingungen (links)

akzeptieren. Anschließend sehen Sie das Suchinterface mit dem ausgewählten TIGER-Korpus und Suchergebnissen für das Lemma »schwellen«. Beachten Sie, dass die Suchanfragesprache hier die des ANNIS-Suchsystems (die ANNIS Query Language) ist und nicht die des TIGERSearch-Systems (s. Kap. 3.1.2.2 f.).

Im TIGERSearch-Interface (s. Abb. 3.5) wird oben links im Fenster ein bereits ausgewähltes Korpus dargestellt, dessen Größe und Annotationen im darunterliegenden Fenster zusammengefasst werden. Im großen Fenster rechts daneben erfolgt die Eingabe der Suchanfrage. Wenn diese mit dem »Search«-Knopf unten rechts abgeschickt wird, öffnet sich ein weiteres Fenster mit Suchergebnissen. Oberhalb des Suchfensters befinden sich Funktionsknöpfe, mit denen das Korpus unabhängig oder abhängig von einer Suchanfrage eingesehen werden kann, Sucheinstellungen modifiziert werden können, Statistiken erstellt oder Treffer exportiert werden können. Diese Funktionen doppeln sich in der obersten Fensterleiste. Verschiedene Funktionen werden weiter unten erläutert.

Das TIGERSearch-Suchinterface verfügt zur Freitexteingabe zusätzlich über einen graphischen Eingabemodus, der direkt über dem Sucheingabefenster angewählt werden kann.

TüNDRA (https://weblicht.sfs.uni-tuebingen.de/Tundra) ist eine Online-Plattform, die die TIGERSearch-Anfragesyntax unterstützt und mit der diverse internationale und deutsche Baumbanken durchsuchbar sind.

Um die online verfügbare Suchoberfläche TüNDRA, dargestellt in Abb. 3.6, erreichen zu können, muss man sich mit Zugangsdaten einer CLARIN-assoziierten Forschungseinrichtung einloggen (in vielen Fällen sind dies die Nutzerdaten des Mediensystems einer Universität). Bevor dann das Suchinterface geöffnet wird, muss eine der in TüNDRA verfügbaren Baumbanken ausgewählt werden. Anschließend kann oben im

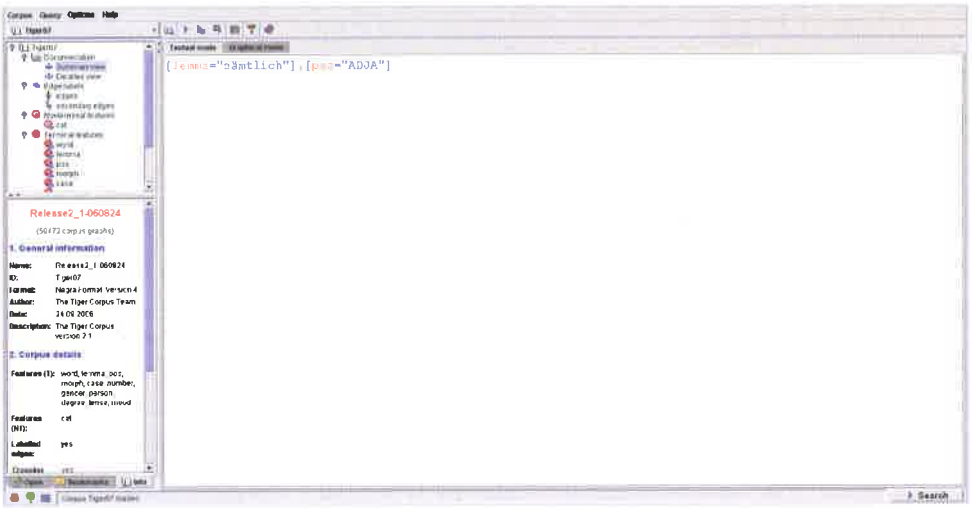


Abb. 3.5: TIGERSearch-Nutzeroberfläche (lokale Installation erhältlich unter https://bit.ly/2TkhTsV)

Abb. 3.6: TüNDRA-Nutzeroberfläche (https://bit.ly/2HOBlfl)

Sucheingabefenster eine Suchanfrage formuliert werden (das System kann die unten erläuterten Suchausdrücke für das Suchprogramm TIGER-Search interpretieren. Mit dem »Run«-Knopf wird die Anfrage abgeschickt. In demselben Browserfenster wird immer genau ein Satz mit einem Treffer angezeigt. Oberhalb des angezeigten Treffers wird die Gesamtzahl der Treffer (Mitte) dargestellt und man kann zwischen den einzelnen Treffern hin- und herspringen. Eine tabellarische Ansicht, Export- und Statistikmöglichkeiten finden sich im unteren Bereich des Browserfensters, wenn man herunterscrollt.

Führen Sie probehalber erste Korpussuchen auf den folgenden registrierungsfrei zugänglichen Suchplattformen durch, um sich mit der Korpussuche vertraut zu machen.

Anleitung

ANNIS

- Wählen Sie die Webadresse https://hu.berlin/annis-intro an.
- Wählen Sie aus der Korpusauswahlliste (links mittig-unten) das Korpus »Parlamentsreden«.
- Geben Sie in das Eingabefenster (links oben) für die Korpussuche den folgenden Suchausdruck ein: `"gehen"`
- Betätigen Sie die Funktion »Search« (unter dem Sucheingabefenster).
- Scrollen Sie durch die Ergebnisse (rechte Seite). Passen Sie ggf. die Größe des linken und rechten Kontexts an (ganz rechts).
- Wählen Sie die Funktion »Search Options« aus, stellen Sie dort den linken und rechten Trefferkontext jeweils auf die Größe 15, führen Sie die Suche (»Search«) noch einmal aus und wiederholen Sie die Trefferansicht.

CQP

- Wählen Sie die Webadresse https://hu.berlin/cqp an. Loggen Sie sich mit den Zugangsdaten »CQP_Demo« (unter »username«) und »Test-Suchen« (unter »password«) ein.

- Wählen Sie aus der Korpusauswahlliste (oben rechts) das Korpus »Parlamentsreden«.
- Geben Sie in das Eingabefenster (oben mittig) für die Korpussuche den folgenden Suchausdruck ein: `"gehen"`
- Betätigen Sie die Funktion »Search« (unten rechts).
- Scrollen Sie durch die Ergebnisse im geöffneten Zusatzfenster.
- Wählen Sie im Hauptfenster bei den Reitern für den linken und rechten Kontext jeweils die Größe 15, führen Sie die Suche (»Search«) noch einmal aus und wiederholen Sie die Treffereinsicht.

TIGERSearch

- Wählen Sie die Webadresse http://fnps.coli.uni-saarland.de:8080/query an.
- Wählen Sie aus der Korpusauswahlliste (mittig-links) das Korpus »TIGER 2.1«.
- Geben Sie in das Eingabefenster (oben mittig) für die Korpussuche den folgenden Suchausdruck ein: `[word="gehen"]`
- Betätigen Sie die Funktion »Evaluate« (mittig-links).
- Skippen Sie durch die Ergebnisse, indem Sie unten bei der Anzeige »Graphs« nach rechts weiterklicken.

Hinweis: Wie oben beschrieben, können Sie zusätzlich die Zugänge zu den Suchinterfaces NoSketch-Engine der COW-Gruppe (http://www.webcorpora.org/) und für TüNDRA (https://bit.ly/2HOBlfl) erlangen und erste Suchen abschicken. Für NoSketch-Engine entspricht der Suchausdruck dem für CQP, für TüNDRA der für TIGERSearch.

3.1.2.1 | Auswahl von Korpusdaten

Alle vorgestellten Suchwerkzeuge können im Grunde beliebig viele Korpora mit verschiedenen Annotationen verwalten. Die Nutzerinnen und Nutzer müssen deshalb grundsätzlich eine Korpusauswahl treffen, bevor Suchanfragen möglich sind. Für die einzelnen Werkzeuge funktioniert dies wie folgt.

ANNIS: Im geöffneten Interface sehen Sie sämtliche verfügbaren Korpora in dem Fenster unten links (s. Abb. 3.2). Hiervon kann ein einzelnes Korpus oder auch mehrere Korpora ausgewählt werden (die mit der linken Maustaste ausgewählten Korpora sind dunkelblau markiert). Die Formulierung einer Suche im Texteingabefeld oben links (die Suche wird mit dem »Search«-Knopf oder mit der Tastenkombination STRG-Return abgeschickt) führt zur Durchsuchung der angewählten Korpora.

CQP und NoSketch Engine: Die online verfügbaren CQP-Suchoberflächen besitzen eine Auswahlmöglichkeit für ein Korpus aus den jeweils verfügbaren. Ohne dass ein bestimmtes Korpus angewählt ist, erfolgt keine Durchsuchung der Ressourcen. Vergleichen Sie beispielhaft Abb. 3.3 für ein CQP-Suchinterface: In dem CQP-Webinterface der Humboldt-Uni-

versität zu Berlin wählt man oben rechts eines der verfügbaren Korpora aus und formuliert dann links in der Eingabeleiste eine Suche.

Im NoSketch-Engine-Interface der Gruppe ›Corpora from the Web‹ (s. Abb. 3.4) zur Durchsuchung der COW-Korpora (Schäfer 2016b) erfolgt die Korpusauswahl ebenso über ein Auswahlmenü (oben).

TIGERSearch: Im TIGERSearch-Interface wird dasjenige Korpus geladen, welches bei der letzten Suche vor dem Schließen des Programms als letztes verwendet wurde. Ist noch kein Korpus geladen, so muss eines der installierten Korpora unter dem Menüpunkt »Corpus« > »Open« angewählt werden (die Korpora erscheinen im Fenster links oben, s. Abb. 3.5). Im mittleren Suchfenster formulierte Suchanfragen beziehen sich auf diese Ressource und werden mit dem »Search«-Knopf unten rechts oder dem Dreiecksymbol oberhalb des Anfragefensters abgeschickt.

TüNDRA: Im online zugänglichen Suchinterface (s. Abb. 3.6) wird ein zuvor ausgewähltes Korpus zur Auswertung bereitgestellt. Um zwischen verschiedenen Korpora zu wechseln, ist also ein erneuter Login ins Interface notwendig.

Bitte diesen Absatz unbedingt hier oder vor TAB 3.1 einfügen

3.1.2.2 | Suche nach Wortformen

Die intuitivste Art, ein Korpus zu durchsuchen, ist die Eingabe konkreter Wortformen. Die Suche liefert die Treffer dann genau gemäß dem eingegeben Ausdruck (in aller Regel betrifft dies auch die Groß- und Kleinschreibung).

Eine Suche nach der Form *geben* wird also Textdaten um den Treffer »*geben*« ausgeben. Im Korpus enthaltene Vorkommen von »*gibst*«, »*gab*«, »*Geben*« usw. werden nicht gefunden. Ebenso wenig wird zwischen finiten Vorkommen von *geben* (in *Sie* ***geben*** *einen Überblick.*) und infiniten Vorkommen (in *Das wird es mit uns nicht* ***geben****.*) differenziert.

Eine Suche nach der Form *gut* findet keine Fälle von »*gute*«, »*besser*«, »*Gut*« oder »*saugut*«, und es wird auch nicht zwischen dem Vorkommen von *gut* als unflektiertes Adjektiv (in *Das ist* ***gut****.*) und als Partikel (in *Es waren* ***gut*** *zwei Kilometer Weg.*) unterschieden.

Eine solche Suche bezieht sich in aller Regel auf den Text, der dem Korpus zugrunde liegt, also auf die Primärdaten. Diese Ebene ist in den meisten Korpora die Token-Ebene, die zugleich die oberste Referenzebene in dem gegebenen Korpus ist. In den Suchprogrammen CQP/NoSketch Engine und TIGERSearch/TüNDRA wird erwartet, dass eine solche tokenisierte Wort-Ebene existiert. Diese trägt immer den Ebenennamen »word« (diese Referenzebene kann also nicht anders benannt werden). In ANNIS kann die wichtigste Referenzebene mit beliebigen Elementen (tokenisierte Wörter, Silben, Zeitabschnitte usw.) versehen werden. Sie muss lediglich die granularste Ebene im Korpus sein (es kann keine feineren Segmentierungen geben) und trägt immer den Namen »tok«. Wenn die »tok«-Ebene in einem ANNIS-Korpus tokenisierten Text enthält, so entspricht die »tok«-Ebene in ANNIS den »word«-Ebenen in Korpora, die in CQP/NoSketch Engine- oder TIGERSearch/TüNDRA gespeichert sind. Suchanfragen für bestimmte Wortformen in solchen Korpora führt Tab. 3.1 auf.

Bitte verwenden Sie, sofern nicht anders angegeben, für die folgenden Suchanweisungen und Aufgaben die genannten Korpora in den entsprechenden Suchinterfaces:
ANNIS (https://hu-berlin.de/annis-intro, bitte Vereinbarung links bestätigen); >>tiger2<<;
CQP (https://hu-berlin.de/cqp, Login: >>CQP_Demo<<, Passwort: >>TestSuchen<<);
>>TIGER Release 2 (CQPified)<<;
TIGERSearch (http://fnms.coli.uni-saarland.de:8080/query/); >>TIGER 2.1<<.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok="geben" (oder) "geben"	[word="geben"] (oder) "geben"	[word="geben"]
Ein gefundener Beispielsatz könnte z. B. sein: *Wir **geben** ganz bestimmt nicht auf.* Achtung: Durch die angegebenen Suchanfragen wird der Fall ***Geben** Sie niemals auf!* Nicht gefunden, weil die Suchsysteme zwischen groß- und kleingeschriebenen Buchstabenzeichen unterscheiden. Man nennt diese Suchfunktion »case sensitive«. Wie man »case insensitive« sucht, wird weiter unten beschrieben. Genauso wenig wird der Fall *Du **gibst** doch jetzt nicht auf!* gefunden, weil die Oberflächenform nicht der gesuchten Form entspricht.		
tok="Gibst" (oder) "gibst"	[word="Gibst"] (oder) "gibst"	[word="Gibst"]
Hierdurch wird z. B. der Fall ***Gibst** du etwas auf?* gefunden, weil der Treffer exakt dem gesuchten Ausdruck entspricht.		
tok="?!" (oder) "?!"	[word="?!"] (oder) "?!"	[word="?!"]
In Kap. 2.2.5 wurde besprochen, dass sämtliche Satzzeichen als eigenständige Funktionseinheiten des Satzes im Korpus systematisch von den Textwörtern abgetrennt werden, damit sowohl Wörter als auch Satzzeichen durch antizipierbare Suchausdrücke gefunden werden können. Dementsprechend werden durch die angegebenen Suchanfragen alle Fälle gefunden, in denen die Tokenisierung aufeinanderfolgende Vorkommen von Frage- und Ausrufezeichen als eine Funktionseinheit segmentiert hat. Gefunden würde z. B. der Fall *Du willst doch nicht etwa aufgeben **?!***		

Tab. 3.1: Beispielsuchanfragen für die Oberflächensuche nach dem Ausdruck »geben«

Arbeitsaufgaben

1. Formulieren Sie Suchausdrücke für die Suche nach der Imperativform des Verbs *halten* in allen drei Anfragesprachen und testen Sie diese Anfragen auf den in Kap. 3.1.2 vorgestellten Suchportalen.
   - Was ist das Problem an der ausgegebenen Treffermenge?
   - Welche Informationen müsste man abfragen, um dieses Problem zu vermeiden?

2. Sie wollen Parenthesen (Einschübe im Satz oder herausgestellte Nachträge) finden. Mit welchen Suchanfragen für die jeweiligen Suchsysteme können Sie dies erreichen?

3.1.2.3 | Reguläre Ausdrücke / Mustersuchen: Variable Zeichen und Zeichenketten

Reguläre Ausdrücke sind Zeichen mit einer bestimmten Suchbedeutung, die nicht dem wörtlichen Zeichen entspricht. Es sind also Suchoperatoren mit einer konkreten Suchanweisung. Beziehen sich Suchoperatoren auf beliebige Zeichen, so wird auch von Wildcards oder Platzhaltern gesprochen.

Definition

Eine Möglichkeit, mit einem Suchausdruck mehrere ähnliche Elemente im Korpus finden zu können, ist die Formulierung von Suchmustern durch beliebige Zeichen und Zeichenketten. Hierzu bedienen wir uns sogenannter regulärer Ausdrücke.

In den Anfragesprachen von ANNIS und TIGERSearch/TüNDRA müssen reguläre Ausdrücke durch Schrägstriche anstatt Anführungszeichen gekennzeichnet werden. Hierdurch weiß das Suchsystem, dass der innerhalb der Schrägstriche befindliche Ausdruck Suchbefehle gemäß der regulären Operatoren enthält.

Der reguläre Operator ».« bedeutet »jedes beliebige Zeichen«. Auf diese Weise findet der ANNIS-Suchausdruck

```
tok=/./
```

nicht nur Punkte, sondern auch alle anderen Token, die aus einem Zeichen bestehen, z. B. »!«, »9« oder »O«. Der einfache Punkt ».« steht nämlich als regulärer Ausdruck für »ein beliebiges Zeichen«. Das heißt, an der Position des Punktes kann jedes im Korpus vorhandene Zeichen inklusive aller Buchstaben, Zahlen und Satzzeichen stehen.

Im Vergleich dazu liefert der ANNIS-Suchausdruck

```
tok="."
```

ausschließlich tokenisierte Punkte, also den Punkt als Satzbeendungszeichen.

Innerhalb komplexerer Suchausdrücke kann der Punkt in den hier behandelten Suchsystemen wie folgt als regulärer Ausdruck angewendet werden.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/.eben/	[word=".eben"]	[word=/.eben/]
Hierdurch werden Vorkommen wie »*geben*«, »*Geben*«, »*Leben*«, »*leben*«, »*beben*«, »*Reben*« usw. gefunden, also jedes beliebige Zeichen, gefolgt von der Zeichenkette »*eben*«.		
tok=/vo./	[word="vo."]	[word=/vo./]
Hierdurch werden Vorkommen wie »*von*«, »*vor*« und »*vom*« gefunden, also jedes beliebige Zeichen, das der Zeichenkette »*vo*« folgt.		

Tab. 3.2: Beispielsuchanfragen für die Verwendung des regulären Ausdrucks ».«

Tab. 3.3: Beispielsuchanfragen für die Verwendung des regulären Ausdrucks »*« und »+«

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/o*h/	[word="o*h"]	[word=/o*h/]
Hierdurch werden Vorkommen wie »*oh*«, »*ooh*«, »*ooooooooooh*« usw. gefunden, aber auch »*h*«.		
tok=/o+h/	[word="o+h"]	[word=/o+h/]
Hierdurch werden Vorkommen wie »*oh*«, »*ooh*«, »*ooooooooooh*« usw. gefunden, aber nicht der Ausdruck »*h*«.		

Man kann beliebig viele Punkte an beliebigen Stellen im Suchausdruck als Variablen bzw. Platzhalter setzen.

Der Asterisk »*« steht für »beliebig oft« und bezieht sich auf das vorangegangene Zeichen.

Das Plus-Symbol »+« steht für »mindestens einmal«. Somit unterscheidet sich die Semantik der Operatoren »*« und »+« genau um den Wert Null (dieser ist in »*« eingeschlossen, in »+« ausgeschlossen).

Die Kombinationen ».*« und ».+« ergeben eine häufig in Korpussuchen erforderte Semantik, nämlich ›ein beliebiges Zeichen beliebig oft‹ bzw. ›ein beliebiges Zeichen mindestens einmal‹. Somit können beliebige Zeichenketten und somit beliebige Wortbestandteile ausgedrückt werden.

Möchten Sie also Wörter finden, die mit »*-barkeit*« enden, so können Sie dies mit ».*barkeit« formulieren:

Tab. 3.4: Beispielsuchanfragen für die Verwendung des regulären Ausdrucks ».*« und ».+«

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/.*barkeit/	[word=".*barkeit"]	[word=/.*barkeit/]
Hierdurch werden Vorkommen wie »*Fruchtbarkeit*«, »*Furchtbarkeit*«, »*Begehbarkeit*« usw. gefunden.		
tok=/hinter.*/	[word="hinter.*"]	[word=/hinter.*/]
Hierdurch werden Vorkommen wie »*hintergehend*«, »*hinterweltlerisch*«, »*hintergründig*« usw. gefunden, aber auch der Ausdruck »*hinter*« selbst. Wenn Sie statt »*« »+« verwenden, wird dies vermieden.		

Arbeitsaufgabe

Formulieren Sie Suchen nach den folgenden Wortformmustern für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Wörter, die mit großgeschriebenem *Z* beginnen.
b) Wörter, die an irgendeiner Stelle den Bestandteil *-bar-* enthalten.
c) Wörter, die an irgendeiner Stelle zwei *e* oder mehr enthalten.

3.1.2.4 | Die Korpussuche als Spezifizierung von Variablen und Werten

Definition

Für ein ideales Verständnis einer Korpus-Mehrebenenarchitektur definieren wir die Begriffe ›Variable‹ und ›Wert‹ wie folgt:
Eine **Variable** ist eine Beschreibungsebene mit variierenden Werten.
Ein **Wert** ist eine bestimmte Interpretation einer linguistischen Einheit.
Diese kann immer einer bestimmten Variable zugeordnet werden.

Bislang haben wir Korpussuchen auf der Token-Ebene durchgeführt, die in den meisten Fällen eine Ebene fortlaufender, tokenisierter Wortformen und Satzzeichen ist. An den bislang verwendeten Suchausdrücken sieht man aber auch, dass genau diese Basisebene durch die Variable »word« oder »tok« beschrieben wird. Die in Anführungszeichen oder Schrägstrichen angegebenen Werte entsprechen Ausdrücken, die auf dieser Korpusebene auftreten können. Es ist notwendig, dass wir bei der Korpussuche immer in solchen Variablen-Wert-Paaren denken: Wenn wir nach Wörtern suchen wollen, die einer bestimmten Wortart, z. B. »NN« für »normales Nomen« gemäß dem STTS, entsprechen, müssen wir zunächst die Beschreibungsebene, also die Variable, angeben, auf der der gesuchte Wert auftreten kann. Dies ist die Beschreibungsebene für Wortarten, die häufig mit »pos« bezeichnet wird. Der Ausdruck, mit dem wir nach normalen Nomen suchen, muss wörtlich übersetzt also lauten »Such auf der Ebene ›pos‹ nach dem Wert ›NN‹.« Siehe Abb. 3.7 zu einer Visualisierung dieser Gegebenheiten.

In den folgenden Kapiteln werden die verschiedenen im Korpus annotierten Phänomene nach dieser Variablen-Wert-Logik gesucht. Bitte beachten Sie dabei stets, dass häufig Standardvariablen wie »pos« für »Wortart« angegeben werden, die aber je nach Korpus auch anders (z. B. »POS« mit Majuskeln oder aber auch »WA« für »Wortart«) bezeichnet werden können.

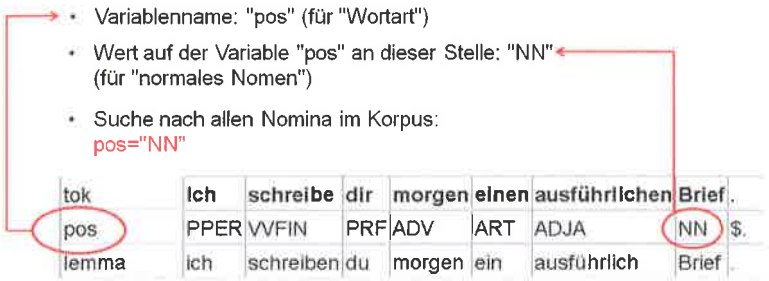


Abb. 3.7: Allgemeine Suchlogik im Sinne der Spezifikation von Variablen und Werten

Arbeitsaufgabe

Hinweis zu der folgenden Aufgabe: Beachten Sie, dass es sich um fingierte Fälle handelt. An dieser Stelle können Sie deshalb ~~nicht wie sonst~~ die formulierten Suchanfragen an authentischen Beispielen ausprobieren.

Stellen Sie sich vor, Sie haben Zugriff auf ein Korpus mit einer Annotationsebene »Exklamation« (Kürzel: EK). Auf dieser Ebene sind exklamative Interjektionen (ITJ), vokativ gebrauchte Nomina (VOK) und imperative Verben (IMP) annotiert. Die Ebene für die fortlaufenden Wortformen bzw. den tokenisierten Text ist mit dem Kürzel TXT (für »Text«) bezeichnet.

a) Wie lautet die Suchanfrage, die in diesem Korpus alle imperativ gebrauchten Verben findet?

b) Wie finden Sie in dem Korpus die Wortform »Mist«?

3.1.2.5 | Suche nach Lemmata

Bei allen flektierbaren Wortarten ist eine Lemmasuche dann relevant, wenn sämtliche Formen eines Flexionsparadigmas gefunden werden sollen: Eine Suche nach dem Lemma *geben* wird nicht nur Kontexte liefern, in denen *geben* auftritt, sondern auch sämtliche flektierte Formen wie *gibst*, *gaben*, *gäbest* usw. ~~finden.~~

Da die Lemmatisierung in einem Korpus immer als eine bestimmte Annotationsebene interpretiert werden muss, ist die Angabe dieser Ebene gemäß der allgemeinen Variablen-Wert-Logik erforderlich. Um nach einem bestimmten Lemma zu suchen, gibt man dem Suchsystem also zunächst den Variablennamen für »Lemma« an (meistens ist dies kleingeschrieben »lemma«) und spezifiziert anschließend einen Wert auf dieser Ebene. Vergleichen Sie die konkreten Suchausdrücke für die verschiedenen Suchsysteme in Tab. 3.5.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
lemma="geben"	[lemma="geben"]	[lemma="geben"]
Hierdurch werden Vorkommen wie »*geben*«, »*gab*«, »*gäbest*« usw. gefunden, also jede beliebige Wortform, die dem Lemma »*geben*« zuzuordnen ist.		
lemma="gut"	[lemma="gut"]	[lemma="gut"]
Hierdurch werden Vorkommen wie »*gut*«, »*gute*« und »*besserem*« gefunden, also jede beliebige Wortform, die dem Lemma »*gut*« zuzuordnen ist.		
lemma=/.*stark/	[lemma=".*stark"]	[lemma=/.*stark/]
Auf sämtlichen Annotationsebenen bzw. Variablen können reguläre Ausdrücke verwendet werden, also auch auf der Lemma-Ebene. Durch die angegebenen Suchausdrücke werden Vorkommen wie »*starkem*«, »*saustarker*« und »*leistungsstärkerem*« gefunden, also jede beliebige Wortform, die einem Lemma zuzuordnen ist, das auf »*-stark*« endet.		

Tab. 3.5: Beispielsuchanfragen für die Lemmasuche

Beachten Sie, dass die Lemma-Ebene im Korpus (sofern überhaupt vorhanden) nicht unbedingt mit dem Variablennamen »lemma« bezeichnet werden muss, sondern dass dieser auch großgeschrieben oder in Versalienschrift geschrieben werden kann oder einen anderen Namen wie »Grundform« tragen kann.

Arbeitsaufgabe

Formulieren Sie Suchen nach den folgenden Lemmata und Lemma-Mustern für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle Wortformen von Wörtern, deren Grundform auf *-lich* endet.

b) Finden Sie alle Wortformen des Flexionsparadigmas, zu dem die Form *wären* gehört.

Hinweis: Sollte eine bestimmte Form eines Paradigmas (wie *wäre*) durch die entsprechende Lemmasuche nicht gefunden werden, so heißt dies nicht, dass die Suche nicht funktioniert, sondern lediglich, dass die entsprechende Form im durchsuchten Korpus nicht enthalten ist.

3.1.2.6 | Suche nach Wortarten

Die Ebene der Wortarten, die wie die Lemma-Ebene meistens mit automatischen Verfahren im Korpus erzeugt wird, trägt meistens den Namen »pos« für Englisch »part of speech« (Wortart). Wenn man nach einem bestimmten Wert für eine Wortart suchen möchte, muss man zunächst die Variable für die Wortartebene angeben und diese mit dem gewünschten Wert verknüpfen (s. Tab. 3.6).

Bitte beachten Sie, dass diese Anfragen nur für Korpora funktionieren, in denen eine Ebene mit der jeweiligen Bezeichnung (»pos« usw.) existiert. Manchmal wird dieser Ausdruck mit Großbuchstaben (»POS«) geschrieben, die Wortartebene kann aber ganz andere Bezeichnungen tragen: Wenn man z. B. das Webinterface für die COW-Korpora (http://www.webcorpora.org/) in der CQL-Einstellung verwendet und eines der DECOW-Versionen durchsucht, muss die Wortart über den Variablennamen »tag« spezifiziert werden. Gegebenenfalls muss man zunächst überprüfen, wie die Ebenen in einem Korpus bezeichnet sind, um dann in der Suche die korrekte Variablenbezeichnung angeben zu können. Die drei bzw. vier hier kontrastierten Suchprogramme verfügen jeweils über Funktionen, anhand derer die in einem bestimmten Korpus vorhandenen Annotationsebenen eingesehen werden können.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
pos="APPR"	[pos="APPR"]	[pos="APPR"]
Hierdurch werden sämtliche Wörter gefunden, die auf der Annotationsebene »pos« den STTS-Wert »APPR« für »Präposition« zugewiesen bekommen haben. Zum Beispiel werden so die Formen »*unter*«, »*trotz*« oder »*auf*« gefunden, sofern sie präpositional verwendet werden und als Präpositionen erkannt wurden.		
pos=/ADJ./	[pos= "ADJ."]	[pos=/ADJ./]
Die Suchanfragen zeigen die Verwendung des regulären Ausdrucks ».« (für »beliebiges Zeichen«) in der Wortartensuche. Hierdurch werden alle Vorkommen gefunden, die auf der Annotationsebene »pos« den Wert »ADJ« und genau ein beliebiges Zeichen besitzen. Dies entspricht mit Blick auf das STTS-Tagset den beiden Tags »ADJA« und »ADJD« für die zwei Adjektivtypen »pränominales Adjektiv« und »prädikatives Adjektiv«. Es werden also Wortformen wie »*gut*« (in *Das ist* ***gut***), »*schlechte*« (in *eine* ***schlechte*** *Note*) und »*erfolgreichem*« (in *manch* ***erfolgreichem*** *Sportler*) gefunden.		
pos=/V.*/	[pos="V.*"]	[pos=/V.*/]
Mit diesen Suchausdrücken werden alle Wörter im Korpus gefunden, die auf der Annotationsebene »pos« einen Wert besitzen, der mit »V« beginnt. Dies entspricht nach dem STTS-Tagset genau allen Verben. Sie erhalten Treffer mit den Oberflächenformen »*Lauf*«, »*gegeben*«, »*wird*« oder »*könntest*«.		

Tab. 3.6: Beispielsuchanfragen für die Wortartensuche

Arbeitsaufgabe

Formulieren Sie Suchen nach den folgenden Wortarten gemäß dem STTS-Tagset (s. Kap. 2.2.6.2) für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle subordinierenden (unterordnenden) Konjunktionen (bzw. Subjunktionen) im Korpus.
b) Finden Sie alle finiten Verben im Korpus.
c) Finden Sie alle Nomina, inklusive Eigennamen, im Korpus.

3.1.2.7 | Suche nach Flexionskategorien

Viele linguistische Fragestellungen erfordern es, dass Wörter in einem bestimmten Flexionsstatus gefunden werden sollen, so z. B. Verben in ihrer Partizipialform (*gefunden*, *gelebt*), Adjektive mit starkem Flexionsgrad (*großer* statt *große* im Nominativ, Singular, Maskulinum) oder nominale Wortarten in bestimmten Kasus.

Solche Flexionseigenschaften, die durch das STTS ausgedrückt werden, können in STTS-getaggten Korpora auf der Wortartenebene gesucht werden. Hierzu zählen die verbalen Eigenschaften »finit« (V.FIN), »in-

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
morph="Dat.Sg.Masc"	[morph="Dat.Sg.Masc"]	[morph="Dat.Sg.Masc"]
Hierdurch werden alle Wörter gefunden, die als Annotationswert genau Dativ Singular Maskulinum (»Dat.Sg.Masc«) besitzen, z. B. »*Hause*« in »*zu Hause*«. Es werden keine Wörter gefunden, die zusätzliche Werte besitzen, wie z. B. Adjektive, deren Flexionsstärke mit ausgewiesen ist.		
morph=/.*Dat.*/	[morph=".*Dat.*"]	[morph=/.*Dat.*/]
Hierdurch werden alle Vorkommen von Wörtern gefunden, bei denen der Dativ mit dem Kürzel »Dat« ausgewiesen ist.		

Tab. 3.7: Beispielsuchanfragen für die flexionsmorphologische Suche unter der Voraussetzung, dass eine Variable »morph« sowie die angegebenen Kürzel im gegebenen Korpus existieren, wie z. B. im TIGER-Korpus der Fall

finit« (V.INF), »partizipial« (V.PP) sowie »imperativisch« (V.IMP) und der Flexionsstatus »flektiert« bzw. »unflektiert« bei Adjektiven (ADJA, ADJD).

Wenn ein Korpus eine Annotationsebene für die Flexionsmorphologie enthält, so kann ihr Name, wie generell bei allen Annotationsebenen, unterschiedlich ausfallen. In den großen Baumbanken TIGER und TüBa-D/Z wird diese Ebene mit »morph« ausgewiesen. Sämtliche Flexionseigenschaften einer gegebenen Wortform sind in einem komplexen Tag konkateniert (verknüpft) und durch Punkte voneinander abgetrennt. Deshalb bietet sich für die Suche nach einzelnen Flexionseigenschaften der reguläre Ausdruck ».*« an (s. Tab. 3.7).

Arbeitsaufgabe

Formulieren Sie Suchen nach den folgenden Flexionskategorien für die einzelnen Suchsysteme. Wählen Sie dazu jeweils das TIGER-Korpus in der entsprechenden Instanz. Die Richtlinien zur Vergabe der flexionsmorphologischen Tags sind in Crysmann et al. 2005 (https://bit.ly/2Hwj9rL) formuliert. Vergleichen Sie auch das Kap. 2.2.6.6 zur Annotation von Flexionsmorphologie. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Wörter im Genitiv.

b) Finden Sie Pronomina und Verben in der ersten Person. (Da die erste Person genau bei Pronomina und Verben annotiert ist, müssen Sie die Suche nicht auf die entsprechenden Wortarten einschränken.)

c) Finden Sie Verben in Präteritalform. (Da die Vergangenheitsstammform ausschließlich auf Verben zutrifft, müssen Sie die Suche nicht auf Verben einschränken.)

3.1.2.8 | Reguläre Ausdrücke / Mustersuchen: Suche nach alternativen Formen oder Zeichenketten innerhalb von Formen

Häufig ist es bei der Korpussuche nötig, Alternativen zu formulieren. Man könnte z. B. daran interessiert sein, alle Elemente im Korpus zu finden, die entweder dem Lemma *Mann*, dem Lemma *Frau* oder dem Lemma *Kind* zugeordnet wurden. Wenn man ein Flexionsparadigma abbilden will, kann man die verschiedenen möglichen Endungen eines flektierbaren Worts als Alternativen angeben. Dann beziehen sich die Alternativen also nicht auf den bzw. die gesamten Suchwert, sondern nur auf einen Teil desselben. Als drittes Szenario möchte man manchmal mehrere Suchanfragen als Alternativen formulieren, um nicht getrennte, sondern eine Trefferliste zu bekommen (s. Tab. 3.8 ganz unten).

Der reguläre Ausdruck, der für »oder« im Sinne einer einschließenden Alternative steht, ist der Pipe (ein vertikaler Strich): »|«. Lesen Sie diesen Operator als einschließendes »oder«. Um den Bereich der als Alternativen zu markierenden Zeichen innerhalb einer Zeichenkette abzustecken, verwendet man runde Klammern, z. B.: »(xyz|zxy)« vs. »xy(z|x)y«. Der erste Suchausdruck findet entweder »xyz« oder »zxy«, der zweite findet entweder »xyzy« oder »xyxy«. Siehe Tab. 3.8 für beispielhafte Alternativensuchen.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
lemma=/(Mann|Frau|Kind)/	[lemma="(Mann|Frau|Kind)"]	[lemma=/(Mann|Frau|Kind)/]
Hierdurch werden sämtliche Wörter gefunden, die auf der Annotationsebene »lemma« entweder die Grundform »*Mann*«, »*Frau*« oder »*Kind*« zugewiesen haben. Mögliche Treffer sind also »*Männern*«, »*Männer*«, »*Mann*«, »*Frauen*«, »*Kinds*«, »*Kindes*« usw.		
tok=/Kind(s|es)/	[word="Kind(s|es)"]	[word=/Kind(s|es)/]
Diese Suchausdrücke finden sowohl die Form »*Kinds*« als auch »*Kindes*«, also beide gültigen Genitivformen von *Kind*.		
pos="ITJ" | tok=/(ähm|äh)/	[pos="ITJ"|word="(ähm|äh)"]	[pos="ITJ" | word=/(ähm|äh)/]
Mit diesen Suchausdrücken werden alle Wörter im Korpus gefunden, die entweder als die Wortart Interjektion (STTS: »ITJ«) ausgewiesen sind, oder die die Oberflächenform »*ähm*« oder »*äh*« besitzen. Elemente, die beide Eigenschaften gleichzeitig besitzen, werden in ANNIS und TIGERSearch/TüNDRA doppelt gefunden, in CQP/NoSketch Engine nur einmal.		

Tab. 3.8: Beispielsuchanfragen für die Suche nach Alternativen

Arbeitsaufgabe

Formulieren Sie Suchen nach den folgenden Alternativen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie durch die Verkettung von Alternativen alle umgelauteten Formen des Verbs *haben* (z. B. *hätten*).
b) Finden Sie durch die Verkettung von Alternativen die STTS-Wortarten Postposition und/oder Zirkumposition.
c) Finden Sie durch die Verkettung von Alternativen die (pronominale) Form *sie* auch am Satzanfang.
d) Finden Sie durch die Verkettung von Alternativen Wortformen, die auf einem Nasal enden.

3.1.2.9 | Reguläre Ausdrücke / Mustersuchen: Mengen von Zeichen an bestimmten Positionen

Die eckigen Klammern in den CQP/NoSketch Engine-Ausdrücken sowie den TIGERSearch- und TüNDRA-Suchausdrücken bezeichnen nicht nur elementare Einheiten im Korpus (Token), sondern können innerhalb der Werteangaben auch Mengen von möglichen Zeichen angeben: Sämtliche als an einer bestimmten Position möglichen Zeichen werden ohne eine bestimmte Ordnung aneinandergereiht. Der folgende Ausdruck gibt an, dass einer von acht möglichen Vokalen auftreten muss (Vorsicht: nur kleingeschriebene Elemente gelten):

```
[eauoüäöy]
```

Mit dem folgenden Ausdruck kann man alle Möglichkeiten für Konsonanten zusammenfassen:

```
[qwrtzpsdfghjklxcvbnmß]
```

»[a-z]« bezeichnet sämtliche Kleinbuchstaben ohne deutsche Sonderzeichen, »[a-zöäüß]« sämtliche Buchstaben unserer Schrift inklusive Umlaute. Großbuchstaben formuliert man analog dazu.

»[0–9]« bezeichnet alle Zahlen von null bis neun. Auf diese Weise können komplexe silbische, morphologische oder mathematische Werte ausgedrückt werden.

Ein hinter der öffnenden Mengenklammer positioniertes ^-Zeichen bezeichnet eine ausgeschlossene Menge; »[^a-z]« findet also alle Zeichen außer die Kleinbuchstaben *a-z*. Vergleichen Sie die Beispiele in Tab. 3.9.

Tab. 3.9: Beispielsuchanfragen für die flexionsmorphologische Suche unter der Voraussetzung, dass eine Variable »morph« sowie die angegebenen Kürzel im gegebenen Korpus existieren, wie z. B. im TIGER-Korpus der Fall

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
lemma=/[nNmMlLjJ].*/	[lemma="[nNmMlLjJ].*"]	[lemma=/[nNmMlLjJ].*/]
Hierdurch werden alle Lemmata gefunden, die auf den sonorsten Konsonanten im Deutschen anlauten.		
lemma=/[A-ZÖÄÜ].*/	[lemma="[A-ZÖÄÜ].*"]	[lemma=/[A-ZÖÄÜ].*/]
Hierdurch werden alle mit Großbuchstaben beginnenden Wörter gefunden.		
tok=/19[0–9][0–9]/	[word="19[0–9][0–9]"]	[word=/19[0–9][0–9]/]
Diese Suchanfragen erfassen sämtliche Jahreszahlen des 20. Jahrhunderts im Korpus.		
lemma=/.*[^aeiouäöüy]/	[lemma=".*[^aeiouäöüy]"]	[lemma=/.*[^aeiouäöüy]/]
Mit diesen Anfragen findet man Lemmata, die nicht vokalisch auslauten. Man findet aber auch Satzzeichen.		

Arbeitsaufgabe

Formulieren Sie Suchen mit Zeichenmengen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle Lemmata, die mit einem Umlaut beginnen.
b) Finden Sie alle Wortformen, die mit einem kleingeschriebenen Vokal beginnen.
c) Finden Sie Wörter, die nicht auf einem Plosiv anlauten.

3.1.2.10 | Reguläre Ausdrücke / Mustersuchen: »Case insensitive«-Suchen

In den vorangegangenen zwei Kapiteln haben Sie bereits das Problem der Groß- und Kleinschreibung von Zeichen bzw. der »case sensitivity« kennengelernt: Groß- und Kleinschreibungen können als Alternativen oder Zweiermengen formuliert werden. In CQP ist aber auch den Operator »%c« anwendbar, der für den gesamten Ausdruck die Groß-/Kleinschreibungsunterscheidung aufhebt. Vergleichen Sie die Möglichkeiten des Umgangs mit variabler Groß- und Kleinschreibung in Tab. 3.10.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/(Ü|ü)ber/ (oder) tok=/[Üü]ber/	[word="(Ü|ü)ber"] (oder) [word="[Üü]ber"]	[word=/(Ü|ü)ber/] (oder) [word=/[Üü]ber/]
Durch die angegebenen Suchanfragen wird gleichermaßen sichergestellt, dass die abstrakte Form *über* im Satzinneren, aber auch großgeschrieben am Satzanfang gefunden werden kann.		
tok=/Bahn(C|c)ard/ (oder) tok=/Bahn[Cc]ard/	[word="Bahn(C|c)ard"] (oder) [word="Bahn[Cc]ard"]	[word=/Bahn(C|c)ard/] (oder) [word=/Bahn[Cc]ard/]
Natürlich kann die Groß-/Kleinschreibungsunterscheidung an jeder Stelle im Wort wie angegeben aufgehoben werden. Die Groß-/Kleinschreibungsunterscheidung kann in CQP aber auch wie folgt generell aufgehoben werden.		
--	[word="über" %c]	--
Hierbei werden sämtliche angegebenen Zeichen nicht nach Groß- und Kleinschreibung unterschieden. Über die Treffermöglichkeiten der oben angegebenen Suchanfragen werden also auch Vorkommen mit Versalienschreibung (*ÜBER*) gefunden.		

Tab. 3.10: Beispielsuchanfragen für die »case insensitive«-Suche

einfache Anf., Bindestrich

Arbeitsaufgabe

Formulieren Sie Suchen mit bestimmten Groß- und Kleinschreibungsanforderungen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Vorkommen der Interjektion *oh* mit beliebig vielen Abfolgen von *o* und *h* sowie möglichst vielen verschiedenen Varianten von Groß- und Kleinschreibung.

b) Finden Sie mit *-in* movierte Nomina im Plural, wobei auch die Variante des Binnen-*I* berücksichtigt wird.

c) Finden Sie gezielt alle Wörter mit Binnenmajuskel (Binnengroßschreibung bzw. Großbuchstaben im Wortinneren).

3.1.2.11 | Reguläre Ausdrücke / Mustersuchen: Suche nach optionalen Elementen

Mit dem regulären Ausdruck »?« werden voranstehende Elemente als optional markiert, d. h. es werden sowohl Ausdrücke gefunden, die den optionalen Teilausdruck enthalten, als auch Ausdrücke, die den optionalen Teilausdruck nicht enthalten. Siehe Tab. 3.11 für Beispiele.

Mit dem ?-Operator ist es in der CQP-Anfragesprache auch möglich, innerhalb von Abfolgen unterschiedlicher Elemente (Wortformen, Lem-

Tab. 3.11: Beispielsuchanfragen für die Suche mit optionalen Elementen

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/Kinde?s/	[word= "Kinde?s"]	[word=/Kinde?s/]
Diese Suchausdrücke finden sowohl die Form »*Kinds*« als auch »*Kindes*«, also beide gültigen Genitivformen von *Kind*.		
tok=/Kind(er)?/	[word="Kind(er)?"]	[word=/Kind(er)?/]
Diese Suchausdrücke finden sowohl die Form »*Kind*« als auch »*Kinder*«, also Nominativ und Akkusativ Singular sowie Nominativ, Genitiv und Akkusativ Plural von *Kind*.		

mata, Wortarten usw.) ein bestimmtes Element als optional innerhalb der Abfolge zu definieren (s. Kap. 3.1.2.15).

Arbeitsaufgabe

Formulieren Sie Suchen mit den folgenden optionalen Elementen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Lemmata, die die Bestandteile *-er*, *-lich* und *-keit* in dieser Reihenfolge enthalten, wobei *-lich* nicht vorkommen muss.

b) Finden Sie Wörter (Wortformen), die auf *-steuer* oder *-steuern* enden und deren Erstbestandteil auf *-t*, *-n* oder *-g* endet. Berücksichtigen Sie, dass zwischen diesen Bestandteilen ein Fugen-*s* stehen kann, aber nicht muss. Die Treffermenge soll beide Varianten enthalten, sofern im Korpus vorhanden.

3.1.2.12 | Reguläre Ausdrücke / Mustersuchen: Interpretation als Operator vs. wörtliches Zeichen

Da die regulären Operatoren, die bisher behandelt wurden, sowohl eine spezifische Bedeutung im System der Operatoren besitzen als auch normale Wort- oder Satzzeichen sein können, muss dem Suchsystem bei der Verwendung der Zeichen angegeben werden, wie die Zeichen (wie z. B. der Punkt) zu interpretieren sind. Im ANNIS-Suchsystem und bei TIGERSearch und TüNDRA kann die Markierung des Suchwerts in Anführungszeichen als wörtlich zu interpretieren angegeben werden; eine Markierung mit Schrägstrichen gibt an, dass als reguläre Ausdrücke interpretierbare Zeichen auch so interpretiert werden sollen. Wenn nun aber ein bestimmtes Zeichen nicht als regulärer Ausdruck interpretiert werden soll, muss dies mit dem Backslash (»\«) angezeigt werden. In CQP gilt dies prinzipiell. Siehe Tab. 3.12 für Beispiele.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
tok=/.*\./	[word= ".*\. "]	[word=/.*\./]
Die im Suchwert angegebenen Zeichen werden teilweise als reguläre Ausdrücke und teilweise als normales Zeichen interpretiert: Während die ersten beiden Zeichen ».*« mit der Bedeutung »eine beliebig lange Zeichenkette« interpretiert werden, wird der Punkt wörtlich genommen, weil ein Backslash vorausgeht. Als Folge werden sämtliche Abkürzungen (wie *usw.*, *Abk.* und *sog.*) gefunden, bei denen auf eine beliebige Zeichenkette als letztes Zeichen der Punkt folgt. Es wird auch die Einheit »...« gefunden.		
tok=/.\?/	[word= ".\?"]	[word=/.\?/]
Diese Suchausdrücke finden ein beliebiges Zeichen, gefolgt von einem Fragezeichen (also z. B. »??« oder »!?«. Wäre das Fragezeichen in der Suchanfrage nicht als wörtlich zu interpretieren markiert, würde es das erste Zeichen des Suchwerts als optional ausweisen.		

Tab. 3.12: Beispielsuchanfragen für die Suche mit ~~komischer~~ Interpretation von regulären Operatoren als reguläre Ausdrücke und wörtlich zu interpretierende Zeichen

Arbeitsaufgabe

Formulieren Sie Suchen mit Zeichen, die als reguläre Ausdrücke interpretiert werden könnten, für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie beliebig viele Abfolgen des Punkts (innerhalb desselben Tokens).
b) Finden Sie mit einem Punkt abgekürzte Wörter.
c) Finden Sie Asteriske (Sternchen) innerhalb der im Korpus verarbeiteten Texte. Diese können entweder alleine stehen oder Teile von Wörtern sein.

3.1.2.13 | Suche nach sämtlichen Werten auf einer bestimmten Annotationsebene

Manchmal möchte man ~~als Korpusnutzerin oder -nutzer~~ sämtliche Vorkommen von einer bestimmten Variablen, unabhängig vom gegebenen Annotationswert, finden. Dies ist z. B. relevant, wenn eine Variablenkategorie nicht für jedes Token relevant ist, sondern an sich nur selten im Korpus auftritt. Stellen Sie sich vor, Sie haben auf einer Annotationsebene »Frage« Entscheidungsfragen (z. B. »EF«) und W-Fragen (z. B. »WF«) ausgezeichnet und wollen zunächst sämtliche Vorkommen auf dieser Ebene, unabhängig von dem vergebenen Wert. Dann lautet der wörtliche Suchbefehl »Finde sämtliche Werte auf der Variable ›Frage‹.« Siehe Tab. 3.13 für Beispiele.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
pos (oder) pos=/.*/	[pos=".*"]	[pos=/.*/]
Hierdurch werden sämtliche Vorkommen von Werten bzw. pos-Tags auf der Ebene »pos« gefunden. Die beiden Anfragen in der ANNIS-Spalte liefern dasselbe Ergebnis; man kann in ANNIS also mit der Angabe der Variable nach sämtlichen Vorkommen auf der entsprechenden Ebene suchen.		

Tab. 3.13: Beispielsuchanfragen für die Suche nach der Variable »pos«

3.1.2.14 | Suche nach mehreren Annotationen bei demselben Token

Sämtliche im Korpus ausgewiesene Annotationen können miteinander Beziehung gesetzt werden. Ein häufiger Fall einer solchen Bezugnahme zwischen verschiedenen Annotationen ist, dass mehrere Eigenschaften auf dasselbe Element im Korpus zutreffen sollen. So kann eine Korpussuche erfordern, dass ein Wort im Korpus ein im Dativ befindliches Adjektiv ist, dass ein Wort das Lemma »unter« trägt und gleichzeitig Präposition (nicht Verbpartikel oder Adjektiv) ist oder dass eine Wortform mit Großbuchstaben beginnt und gleichzeitig ein Adjektiv ist (bei Ortsadjektiven oder Satzanfängen).

Die wörtliche Suchanweisung lautet dabei, dass zwei (oder mehr) Eigenschaften auf dasselbe Element (häufig dasselbe Token) im Korpus zutreffen sollen.

Stellen Sie sich vor, Sie wollen die Form *an* finden, aber ausschließlich in ihrer Verwendung als Präposition, nicht als Verbpartikel oder Adjektiv. Es ist sinnvoll, die Form *an* dann nicht als Oberflächenform auf der Text- bzw. Tokenebene zu suchen, sondern als Lemma, denn im Kontext von Satzanfängen tritt die Form auch großgeschrieben auf. Eine Suchbedingung ist also, dass auf der Annotationsebene der Lemmata die Form »*an*« gesucht wird. Gleichzeitig soll das gesuchte Element auf der Annotationsebene der Wortarten (»pos«), den Wert für Präposition besitzen (nach dem STTS-Tagset »APPR«). In den drei Suchanfragesprachen (s. Tab. 3.14) formuliert man diese zwei verknüpften Bedingungen wie folgt.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
lemma="an" _=_ pos="APPR"	[lemma="an" & pos="APPR"]	[lemma="an" & pos="APPR"]
Die Verknüpfung der zwei Suchbedingungen wird in AnnisQL (der ANNIS-spezifischen Anfragesprache) anders als in den beiden anderen Anfragesprachen ausgedrückt: Der »_=_«-Operator in ANNIS drückt aus, dass die beiden verknüpften Restriktionen in einem Abdeckungsverhältnis zueinander stehen, d. h. sie gelten gleichzeitig an derselben Stelle im Korpus. Durch die Klammern in der CQP/NoSketch Engine- und der TIGERSearch- bzw. TüNDRA-Anfragesprache wird ausgedrückt, dass es sich um ein Tokenelement im Korpus handelt, der &-Operator drückt aus, dass die zwei Bedingungen gleichzeitig bzw. additiv gelten. Die jeweiligen Suchen führen zum selben Ergebnis.		

Tab. 3.14: Beispielsuchanfrage für die Verknüpfung zweier Anforderungen

Auf diese Weise kann man beliebige und beliebig viele (solange es die Rechenkapazität nicht überschreitet) Beschränkungen miteinander verknüpfen.

Arbeitsaufgabe

Formulieren Sie Suchen nach mehreren Merkmalen bei demselben Token für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Vorkommen des Worts (Lemmas) *an*, die abgetrennte Verbpartikeln sind.
b) Finden Sie Vorkommen von *meinen*, die Possessivartikel (nach STTS: attributive Possessivpronomen) sind.
c) Finden Sie Vorkommen von *einen*, die Artikel sind.
d) Finden Sie Vorkommen von *meinen* und *einen*, die Verben sind.

3.1.2.15 | Suche nach Abfolgen

Will man bestimmte syntaktische Strukturen finden und kann mangels echter syntaktischer Annotationen nicht direkt nach Phrasenkategorien und anderen syntaktischen Konstruktionen suchen, so hilft häufig die Möglichkeit, nach Abfolgen bestimmter Wortarten oder Lemmata zu suchen. Die Abfolge wird auch als ›Präzedenz‹ bezeichnet (und meint dabei linguistisch eine spezifisch gerichtete Adjazenz bzw. Kookkurrenz, vgl. Kap. 4.6.2). Ketten von Elementen werden auch ›N-Gramme‹ genannt.

In AnnisQL und der TIGERSearch- bzw. TüNDRA-Anfragesyntax werden unmittelbare Abfolgen mit dem Punkt (».«) zwischen zwei Elementen ausgedrückt. Somit ist der Punkt als Suchoperator doppeldeutig: Er steht sowohl für »ein beliebiges Zeichen« als auch für unmittelbare Präzedenz. Durch eine Zahl nach dem Punkt kann auch ein beliebiger Abstand zwischen den präzedenten Elementen definiert werden (z. B. bezeichnet ».4« einen Abstand von vier Stellen).

In CQP/NoSketch Engine werden anstelle eines Abstandsoperators einfach zwei (oder mehr) durch eckige Klammern ausgedrückte Elementareinheiten nebeneinandergesetzt.

Bei beiden Varianten der Abstandsformulierung können auch Abstandsbereiche angegeben werden (z. B. »im Bereich von zwei bis vier Stellen«).

~~Somit sind die Suchausdrücke in Tab. 3.15 synonym.~~

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
pos="APPR" . pos="NN"	[pos="APPR"] [pos="NN"]	[pos="APPR"] . [pos="NN"]
Hierdurch werden sämtliche Vorkommen gefunden, in denen auf eine Präposition unmittelbar ein Nomen folgt (z. B. ~~bei~~ *»durch Übermüdung«*). In ANNIS und TIGERSearch und TüNDRA drückt der Punkt zwischen zwei Suchelementen also unmittelbare Abfolge aus, während in CQP unmittelbar präzedente Elemente einfach durch ein Nebeneinandersetzen (Leerzeichen können, müssen aber nicht gesetzt werden) ausgedrückt werden.		
pos="APPR" .2 pos="NN"	[pos="APPR"] [] [pos="NN"]	[pos="APPR"] .2 [pos="NN"]
Hierdurch werden sämtliche Vorkommen gefunden, in denen auf eine Präposition in einem Abstand von zwei Stellen ein Nomen folgt (z. B. ~~bei~~ *»durch reine Übermüdung«*). In CQP/NoSketch Engine wird dies erreicht, indem durch eine leere eckige Klammer ein beliebiges Element zwischen Präposition und Nomen gesetzt wird.		
pos="APPR" .2,4 pos="NN"	[pos="APPR"] []{1,3} [pos="NN"]	[pos="APPR"] .2,4 [pos="NN"]
Hierdurch werden sämtliche Vorkommen gefunden, in denen auf eine Präposition in einem Abstand von zwei bis vier Stellen ein Nomen folgt (z. B. werden gefunden: *»durch reine Übermüdung«*, *»durch die reinste Übermüdung«*, *»wegen einer sehr großen Sache«*).		
pos="APPR" . pos="ADJA" . pos="NN"	[pos="APPR"] [pos="ADJA"] [pos="NN"]	[pos="APPR"] . #1:[pos="ADJA"] & #1 . [pos="NN"]
Die jeweiligen Suchanfragen finden Sequenzen unmittelbarer Abfolgen von Präposition, Adjektiv und Nomen (z. B. *wegen großer Unsicherheit*). In TIGERSearch ~~bzw. TüNDRA~~ kann jeweils nur eine Relation zwischen zwei Elementen erfragt werden; weitere Relationen müssen separat ausgedrückt werden, indem das relevante Element durch einen Variablennamen wiederaufgenommen wird.		
--	[pos="(P.*AT|ART)"] [pos="ADJA"]* [pos="NN"]	--
In CQP kann man durch den Sternchen-Operator ausdrücken, dass ein bestimmtes Element beliebig häufig auftreten darf. Der abgebildete Suchausdruck findet somit sämtliche Abfolgen von einem Artikelwort (attribuierende Pronomina oder Artikel laut STTS), beliebig vielen Adjektiven ~~ohne aufzählendes Komma~~ und einem Nomen. In den anderen Suchsystemen müssten getrennte Suchen für verschieden viele Adjektive formuliert werden.		
--	[pos="(P.*AT|ART)"] [pos="ADJA"]? [pos="NN"]	--
Ebenso kann man in der CQP-Anfragesprache ein beliebiges Element einer Sequenz (wie hier das PP-interne Adjektiv) optional setzen. Man findet also Abfolgen von Präpositionen und Nomina sowie Abfolgen von Präpositionen, Adjektiven und Nomina. Auch hier müssen in anderen Suchsystemen (zwei) getrennte Anfragen (oder eine Anfrage mit zwei durch den |-Operator getrennten Teilanfragen) gestellt werden.		

Tab. 3.15: Beispielsuchanfragen für die Suche nach Abfolgen

Arbeitsaufgabe

Formulieren Sie Suchen nach Abfolgen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Abfolgen von Pronomina und Nomina.
b) Finden Sie die Negationspartikel *nicht* am Satzende (also vor einem Satzbeendungszeichen).
c) Finden Sie die Konjunktion *und* direkt nach Kommata.
d) Finden Sie Abfolgen des Lemmas *auf*, des Lemmas *jeder* oder *kein* und eines beliebigen Nomens.

3.1.2.16 | Reguläre Ausdrücke / Mustersuchen: Suche mit Negation

In manchen Suchszenarien möchte man bestimmte Elemente ausschließen. Es ist z. B. möglich, dass bestimmte Annotationswerte oder bestimmte innerhalb eines Werts bestimmte Muster ausgeschlossen werden sollen. Als Negationsoperator dient zum einen das Ausrufezeichen, das zwischen eine Variable und dem folgenden Gleichheitszeichen gesetzt werden kann. Zum anderen können durch negative Mengen bestimmte Muster an Werten ausgeschlossen werden. Dabei verwendet man die bereits eingeführten Mengenklammern (eckige Klammern) und stellt das »^«-Symbol vornan, welches man mit »außer« übersetzen kann.

Vergleichen Sie die beispielhaften Ausdrücke in Tab. 3.16 für die Verwendung von Negationsausdrücken in der Korpussuche.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
pos!=/(ART|ADJA)/ . pos="NN"	[pos!= "(ART|ADJA)"] . [pos="NN"]	[pos!=/(ART|ADJA)/] . [pos="NN"]
Hierdurch wird ausgeschlossen, dass vor dem Nomen ein Artikel oder ein Adjektiv steht. Alle anderen Wortarten des STTS sind vor dem Nomen zugelassen.		
lemma="zu" _=_ pos!="APPR"	[lemma="zu" & pos!= "APPR"]	[lemma="zu" & pos!= "APPR"]
Hierdurch wird das Lemma »*zu*« gefunden (also die Wortformen »*zu*« und »*Zu*« im Text), wenn sie nicht als Präposition (sondern z. B. als Verbpartikel) verwendet werden.		
tok=/[^aeuioüöäy].*/	[word="[^aeuioüöäy].*"]	[word=/[^aeuioüöäy].*/]
Hierdurch werden sämtliche Wortformen gefunden, die nicht mit den kleingeschriebenen Vokalen [illegible] die abgebildet sind, beginnen.		
tok=/[^0–9]*/	[word="[^0–9]*"]	word=/[^0–9]*/]
Hierdurch werden alle Zeichenketten gefunden, die keine Zahlen enthalten. Die wörtliche Übersetzung des Suchausdrucks ist jeweils: »Finde auf der Ebene ›tok‹ bzw. ›word‹ beliebig viele aufeinanderfolgende Zeichen, die jeweils nicht die Zahlen von null bis neun sind.«		

Tab. 3.16: Beispielsuchanfragen für die Suche nach ausgeschlossenen Elementen

Arbeitsaufgabe

Formulieren Sie Suchen nach ausgeschlossenen Elementen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Lemmata, die auf *-en* enden, aber keine Verben sind.
b) Finden Sie Wortformen, die kein *e* oder *E* enthalten.
c) Finden Sie Abfolgen von Artikel (nach STTS-Definition), einem Element und Nomen, wobei das mittlere Element kein Adjektiv sein soll.

3.1.2.17 | Zusammenfassung der behandelten regulären Ausdrücke

Zur besseren Übersicht sind in der nachfolgenden Zusammenstellung die bislang behandelten regulären Ausdrücke aufgelistet.

Zeichen	Bedeutung
.	An dieser Stelle steht ein beliebiges Zeichen.
*	Das vorangegangenen Element gilt beliebig oft (inklusive null Vorkommen).
.*	Hier steht eine beliebig lange Zeichenkette mit allen möglichen Zeichen (in beliebiger Abfolge).
+	Das vorangegangene Element tritt mindestens einmal auf.
?	Das vorherige Element ist optional.
\	Das folgende Zeichen wird wörtlich genommen.
!	Der folgende Wert gilt nicht, sondern jeder andere.
[abc]	An dieser Stelle gilt ein Zeichen der Menge a, b und c.
[^abc]	An dieser Stelle gilt jedes Zeichen außer dieser Menge.
(a|b)	An dieser Stelle gelten genau a oder b.

3.1.2.18 | Zusammenfassung der Ebenen, die in Standardkorpora durchsucht werden können

Normalerweise lassen sich Korpora tokengenau auf den folgenden Ebenen durchsuchen:

- Wortform (und Wortformmuster mittels regulärer Ausdrücke)
- Lemma (und entsprechende Muster)
- Wortart (sehr häufig nach dem STTS-System)

Diese Ebenen lassen sich als Suchvariablen wie folgt in Beziehung setzen:

- in einem Abdeckungsverhältnis bzw. als gebündelte Merkmale bei demselben Token
- als unmittelbare oder mittelbare Abfolgen

~~Weitere, spezifischere Annotationen und Relationen zwischen Suchvariablen werden in den kommenden Kapiteln behandelt.~~

Mit diesem Inventar an Möglichkeiten lassen sich bereits sehr spezifische Syntagmen bzw. Konstruktionen suchen. Es lassen sich z. B. Satzanfänge und Satzenden definieren, indem Elemente nach bzw. vor satzbeendender Interpunktion gesucht werden. Man kann auch bestimmte Vorfeld- oder Mittelfeldkonstituenten definieren. Siehe hierzu die folgende Arbeitsaufgaben.

Arbeitsaufgaben

1. Formulieren Sie für die CQP-Anfragesprache eine Suche nach einer Präpositionalphrase mit postnominalem (nachgestelltem) *wegen* im Vorfeld eines Satzes, indem Sie in der Anfrage die folgende Abfolge festlegen: Satzbeendungszeichen – optionales Artikelwort (»ART« oder »P.*AT« nach STTS) – optionales Adjektiv – Nomen (»NN« oder »NE« nach STTS) – finites Verb.
~~1.~~ Formulieren Sie analog dazu eine Suchanfrage mit präpositionalem (vorangestelltem) *wegen*.
Führen Sie die Suchen für das Korpus »DeWaC 1« im CQP-Interface unter http://~~korpling.german~~hu-berlin.de/cqp~~/~~ durch und vergleichen Sie die Treffer für die unterschiedlichen Anfragen. Vergleichen Sie Arbeitsaufgabe 3 in Kapitel 4.6.1 für eine Auswertung der Suchergebnisse.

3.1.2.19 | Suche nach Relationen zwischen Token und Spannen

Wenn man mit sogenannten Partitureditoren wie EXMARaLDA (s. z. B. Kap. 2.4.1) oder Excel annotiert hat, um z. B. Satzspannen, topologische Felder oder andere linguistische Konzepte zu annotieren, die sich ideal

tok	Ich	schreibe	dir	morgen	einen	ausführlichen	Brief	.
pos	PPER	VVFIN	PRF	ADV	ART	ADJA	NN	$.
lemma	ich	schreiben	du	morgen	ein	ausführlich	Brief	.
Satz	S							
TopFeld	VF	LSK	MF					
vp		vp						
Satzfunktion	SUBJ	PrädFin	OBJD	ADV	OBJA			
Definitiheit	DEF		DEF		INDEF			

Abb. 3.8: Partitur eines Satzes mit verschiedenen Spannenannotationen

als Spannen abbilden lassen, so ist es nicht nur möglich, die annotierten Spannen selber zu suchen, sondern alle anderen Informationen im Korpus können mit der Spannenannotation in Beziehung gesetzt werden.

Der in Abb. 3.8 analysierte Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* ist auf sieben Annotationsebenen linguistisch annotiert: »pos« steht für »Wortart« und enthält als Werte die STTS-Wortarten, »lemma« steht für »Lemma« und enthält als Werte die Grundformen der Wortformen des Beispielsatzes, »Satz« chunkt die Sätze zu Spannen mit dem Wert »S«, »TopFeld« steht für »topologische Felder« und enthält die Werte »VF« für »Vorfeld«, »LSK« für »linke Satzklammer« und »MF« für »Mittelfeld«, »vp« chunkt mit dem Wert »vp« das Vollverb mit seinen Argumenten und Adjunkten zu einer Spanne, auf der Ebene »Satzfunktion« befinden sich die Werte »SUBJ« für »Subjekt«, »PrädFin« für »finiter Prädikatsteil«, »OBJD« für »Dativobjekt«, »ADV« für »Adverbial« und »OBJA« für »Akkusativobjekt« und auf der Ebene »Definitheit« sind die beiden Personalpronomina mit »DEF« für »definit« und die Akkusativobjekts-NP mit »INDEF« für »indefinit« gekennzeichnet.

Man kann solche Strukturen systematisch durchsuchen: Jede erdenkliche Beziehung zwischen den Token und annotierten Spannen kann in der Suche berücksichtigt werden. Hierfür werden bestimmte Operatoren verwendet, die genau diese Relationen ausdrücken. Die Aufstellung in Tab. 3.17 soll die Möglichkeiten umfassend aufzeigen. Da ANNIS dasjenige Suchwerkzeug ist, welches mit dem Spannenkonzept ohne starke Beschränkungen (wie bei den anderen Werkzeugen der Fall) umgehen kann, werden ausschließlich für dieses Werkzeug Suchlösungen formuliert. Die Suchanfragen beziehen sich auf die Variablen- und Wertbezeichnungen, die in Abb. 3.8 aufgeführt sind. Sämtliche Zellen, auch die Token-Zellen in der obersten Zeile, werden als derselbe Typ von Spanne behandelt (s. Tab. 3.17).

Wenn man mehrere Beziehungen, z. B. die Abfolge und die Abdeckung, in einer Suchanfrage vereinen möchte, muss man den gesuchten Elementen Variablennamen geben, um mehrfach in der Suchanfrage auf sie verweisen zu können. So kann man z. B. auf die Struktur in Abb. 3.8 ausdrücken, dass man die Abfolge von Dativobjekt und Adverbial sucht, und die beiden Elemente innerhalb desselben Mittelfelds auftreten sollen. Wie man die nötigen Verweise in der Suchanfrage herstellt, wird ab Kapitel 3.1.2.21 beschrieben.

Arbeitsaufgaben

1. Die folgenden Suchanfragen beziehen sich zunächst auf die Spannen-Annotationen in der Abb. 3.8. Nutzen Sie die dort abgebildeten Variablen und Werte. Sie können diese Suchanfragen nicht überprüfen, weil das Korpusbeispiel fingiert ist.
   a) Finden Sie alle Fälle, in denen (wie in der Abbildung) nach der linken Satzklammer unmittelbar ein Dativobjekt folgt.
   b) Finden Sie alle Fälle, in denen (wie in der Abbildung) das Mittelfeld ein Adverbial enthält.

ANNIS	CQP/NoSketch Engine	TIGERSearch/ TüNDRA
TopFeld="MF"	--	--
Diese Suche mit einem Variable-Wert-Ausdruck ist das einfachste Suchszenario: Es wird nach einer Zelle eines bestimmten Typs auf einer bestimmten Annotationsebene gesucht. Diese Suchanfrage findet alle auf der Ebene »TopFeld« (für »topologische Felder«) als »MF« (für »Mittelfeld«) ausgewiesene Zellen bzw. Spannen.		
TopFeld="LSK" _=_ Satzfunktion="PrädFin"	--	--
Der Abdeckungsoperator »_=_« wurde bereits vorgestellt. Er gilt nicht nur für Tokenelemente, sondern genauso für Spannen. Dementsprechend findet die Suche sämtliche Vorkommen, bei denen ein auf der Annotationsebene »TopFeld« als »LSK« (»linke Satzklammer«) ausgewiesenes Element genau mit einem auf der Ebene »Satzfunktion« als »PrädFin« (»finiter Prädikatsteil«) ausgewiesenes Element abdeckt. Entsprechend können beliebig große Spannen dabei berücksichtigt werden.		
TopFeld="MF" _i_ Satzfunktion="OBJD"	--	--
Mit dem Operator »_i_« wird Inklusion ausgedrückt: Die abgebildete Suchanfrage findet sämtliche Vorkommen, in denen ein auf der Ebene »Satzfunktion« als »OBJD« (»Dativobjekt«) ausgewiesenes Element in einer Spanne mit der Bezeichnung »MF« (»Mittelfeld«) auf der Annotationsebene »TopFeld« (für »topologische Felder«) enthalten ist. Abdeckende Spannen werden mit dieser Suchoperation auch gefunden, der Inklusionsoperator findet also eine Obermenge des Abdeckungsoperators.		
TopFeld="MF" _l_ Satzfunktion="OBJD"	--	--
Der Operator »_l_« steht für Links-Alignierung: Im Gegensatz zur vorigen Anfrage werden hier nur »OBJD«-Elemente gefunden, die links-aligniert, also links-abschließend mit der sie enthaltenen »MF«-Spanne sind. Der abgebildete Fall wird also gefunden. Der Links-Alignierungsoperator (wie der nachfolgende Rechts-Alignierungsoperator) findet eine Untermenge der »_i_«-Beziehungen.		
Satz _r_ Satzfunktion="OBJA"	--	--
Der Operator »_r_« steht für Rechts-Alignierung: Die Suchanfrage findet sämtliche auf der Ebene »Satz« verzeichneten Spannen (unabhängig von deren Benennung), die rechts-alignierend eine Spanne mit der Bezeichnung »OBJA« (für »Akkusativobjekt«) auf der Annotationsebene »Satzfunktion« enthalten.		
TopFeld="LSK" _o_ vp	--	--
Der am seltensten verwendete Spannenrelationsoperator ist der Überlappungsoperator »_o_«. Er findet überlappende Spannen wie die in der Suche angegebenen Spannen »MF« (für »Mittelfeld«) auf der Ebene »TopFeld« und die »vp«-Spanne (für »kleine vp«).		
pos="VVFIN" . Satzfunktion="OBJD"	--	--
Diese Anfrage illustriert, dass direkte (und genauso indirekte Abfolgen) auch auf Spannen auf beliebigen Annotationsebenen bezogen werden können: Gefunden werden alle Fälle, in denen eine auf der Ebene »Satzfunktion« als »OBJD« (für »Dativobjekt«) ausgewiesene Spanne unmittelbar auf eine auf der Ebene »pos« (für »Wortart«) Spanne mit dem Wert »VVFIN« (für »finites Vollverb«) folgt.		

Tab. 3.17: Beispielsuchanfragen für die Suche nach Spannen, Relationen zwischen verschiedenen Spannen sowie zwischen Spannen und Token. Die Anfragen beziehen sich auf die Annotationen in der Abb. 3.8.

c) Finden Sie entgegen der Abbildung alle Fälle, in denen das Mittelfeld nur genau ein Adverbial enthält, also mit ihm deckungsgleich ist. Welche der oben vorgestellten Operatoren, die Sie zwischen die Suchelemente setzen können, finden diese Fälle auch (und zusätzlich weitere)?

2. Die folgenden Suchaufgaben beziehen sich auf das BeMaTaC-Korpus BeMaTaC_L1_3.0 in der ANNIS-Instanz https://korpling.germanistik.hu-berlin.de/annis3/intro. Dieses Korpus besteht aus transkribierten gesprochenen Dialogen, die mit diversen Annotationen angereichert wurden. Entnehmen Sie die für die Suche notwendigen Variablen- und Wertenamen den Aufgaben, formulieren Sie die jeweilige Suche und überprüfen Sie die Suche anhand des BeMaTaC-Korpus.
In dem Korpus sind Äußerungen auf einer Annotationsebene »utt« (für »utterance«) und als Spannen mit der Bezeichnung »utt« annotiert. Die transkribierten und normalisierten Wortformen sind wie in den bisher behandelten Korpora mit Lemma- (Ebene: »lemma«) und Wortartenannotationen (Ebene: »pos«; STTS-Tagset) versehen.

a) Finden Sie äußerungsinitiale Verben (Verben, die am Anfang einer »utt«-Spanne stehen).

Hinweis: Stellen Sie vor dem Abschicken der Suche den linken Trefferkontext (unter dem Reiter »Search Options« in ANNIS) auf null, um eine gute Trefferansicht zu erhalten.

b) Finden Sie Formen des Lemmas *gehen* am Ende von Äußerungen.

Hinweis: Stellen Sie vor dem Abschicken der Suche den linken Trefferkontext auf fünf und den rechten auf null, um eine gute Trefferansicht zu erhalten.

3.1.2.20 | Phrasenstrukturbaumbanken: Suche nach Konstituenten und Phrasen im Satz

Wie in Kapitel 2.2.7 beschrieben, enthalten Konstituenten- bzw. Phrasenstrukturannotationen Informationen über die syntaktischen Phrasen und deren Relationen und Funktionen im Satz. Abb. 3.9 zeigt den bereits behandelten Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* im TIGER-Annotationsformat.

Zusätzlich zu den bisher besprochenen Suchmöglichkeiten lässt sich für Korpora wie das TIGER-Korpus die abgebildete syntaktische Struktur

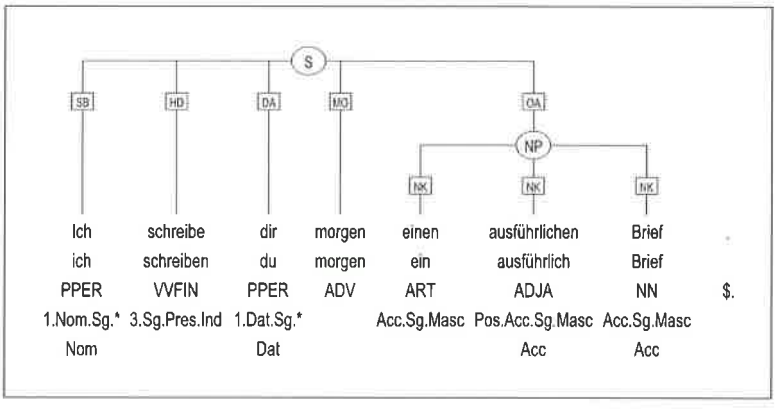


Abb. 3.9: Der Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* im TIGER-Annotationsformat.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
cat="AP"	–	[cat="AP"]
Hierdurch werden sämtliche als Adjektivphrasen ausgezeichnete Strukturen gefunden.		

Tab. 3.18: Beispielsuchanfragen für die Suche nach Phrasenkategorien

in die Suche einbeziehen, indem sich die syntaktischen Phrasenkategorien (die Knoten in der Baumstruktur) und die syntaktischen Funktionen (die Kanten in der Baumstruktur) suchen lassen oder zueinander und zu den übrigen Annotationen in Beziehung setzen lassen.

Die einfachste Möglichkeit der Auswertung solcher Baumbanken ist die Suche nach den in der Baumbank vergebenen Phrasenkategorien. So sollte eine Suche nach der Kategorie »AP« für »Adjektivphrase« auf der korrekten Suchebene bzw. -variable sämtliche Sätze ausgeben, in denen Adjektivphrasen annotiert wurden. Die in Tab. 3.18 dargestellten Korpussuchen beziehen sich auf Korpora, die im Format des TIGER-Annotationsschemas (Albert et al. 2003) erstellt wurden, wie das TIGER-Korpus, das von den existierenden Baumbanken des Deutschen am leichtesten beziehbar ist (für Hinweise zur Nutzung des Korpus s. Kap. 3.1.2). Vergleichen Sie bitte auch das TIGER-Annotationsschema Abert et al. (2003), in dem die Annotationen beschrieben sind (http://www.linguistics.ruhr-uni-bochum.de/~dipper/pub/tiger_annot.pdf); eine Liste der dort definierten Phrasen- und Kantenbezeichnungen finden Sie hier: https://sourceforge.net/projects/lehrbuchkorpling/files/Handreichungen/Tiger_edges_phrases.zip). Die Variable für die Phrasenkategorien im Korpus trägt den Namen »cat«, somit müssen Phrasenwerte wie z. B. »PP« für »Präpositionalphrase« auf dieser Ebene gesucht werden. Weitere wesentliche Phrasenkategorien sind »NP« für »Nominalphrase«, »AP« für »Adjektivphrase«, »AVP« für »Adverbialphrase«, »VP« für »Verbalphrase« und »S« für »Satz«. Exemplarische Suchausdrücke für diese Kategorien finden Sie in Tab. 3.18. Die Anfragesprache von CQP/NoSketch Engine ist dabei nicht berücksichtigt, weil Phrasenbaumstrukturen in CQP/NoSketch Engine nicht abgebildet werden können.

Beachten Sie, dass diese Suchanfragen nur funktionieren, wenn ein gegebenes Korpus die Variable »cat« und den dort annotierten Wert »AP« enthält. Da diese Auszeichnungen spezifisch für das TIGER-Korpus sind, werden die Suchanfragen auch nur dort Ergebnisse erzielen.

Arbeitsaufgabe

Verwenden Sie die oben genannten Informationen zum TIGER-Korpus und formulieren Sie Suchen für das TIGER-Korpus, wenn es im ANNIS-Suchinterface durchsucht wird sowie wenn es in TIGERSearch durchsucht. Sie können für Anfragen in der TIGERSearch-Anfragesprache auch

die Online-Suchinstanz unter der Webadresse http://fnps.coli.uni-saarland.de:8080/ verwenden.

a) Finden Sie alle Präpositionalphrasen.
b) Finden Sie alle koordinierten Sätze.

3.1.2.21 | Phrasenstrukturbaumbanken: Suche nach syntaktischen Relationen

Interessanter werden die Suchmöglichkeiten, wenn man die Suche um relationale Aspekte erweitert, indem man Phrasenkategorien und/oder Wörter, Lemmata und Wortarten zueinander in Beziehung setzt. Hierdurch kann man z. B. bestimmte Phrasen suchen, die in anderen Phrasen enthalten sind (etwa eine Präpositionalphrase, die in einer Verbalphrase auftritt). Hierzu benötigt man den sogenannten Dominanzoperator, der häufig mit der schließenden Spitzklammer (»>«) belegt ist. Wenn zwei Elemente mit dem Dominanzoperator verknüpft sind, so bedeutet dies, dass das erste Element das zweite (direkt) dominiert. Indirekte Dominanz (also Dominanz über mehrere Schritte) drückt man über Zahlen (die jeweilige Zahl bestimmt den Grad der Abhängigkeit) oder den Stern-Operator »*« (dieser bezeichnet einen beliebigen Grad der Abhängigkeit) aus. Dominanzverhältnisse müssen in Phrasenstrukturbäumen nicht zwischen Phrasen bestehen, sondern können auch zwischen übergeordneten Phrasen und untergeordneten Wörtern bestehen. Siehe hierzu Tab. 3.19.

Tab. 3.19: Beispielsuchanfragen für die Suche nach Dominanzbeziehungen zwischen Knoten

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
cat="VP" > cat="PP"	--	[cat="VP"] > [cat="PP"]
Hierdurch werden alle VPn gefunden, die unmittelbar eine PP enthalten.		
cat="PP" >* lemma="sehr"	--	[cat="PP"] >* [lemma="sehr"]
Hierdurch werden alle PPn gefunden, die ein Token mit der Lemma-Annotation »*sehr*« enthalten, unabhängig vom Grad der Abhängigkeit.		

Sobald an eine Phrase mehrere Dominanzanforderungen gestellt sind, wird die Suchanfrage deutlich komplexer. Nehmen wir an, wir möchten alle VPn finden, die eine PP und eine NP enthalten. Dann ist es nötig, dass wir die VP in der Suche referenzieren, denn es soll dieselbe VP sein, die eine PP und eine NP als Tochter hat. In ANNIS erreicht man dies, indem man das entsprechende Element mit »#« und der Ziffer, die der Position des Elements in der Suchanfrage entspricht, aufnimmt. In der TIGERSearch-Anfragesprache werden die zu referenzierenden Elemente mit einem Variablennamen versehen, der dann wiederaufgegriffen wird. Die verschiedenen Teile der Suchanfrage werden durch »&«-Zeichen miteinander verknüpft. Vergleichen Sie die Beispiele in der folgenden Auflistung.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
cat="VP" > cat="PP" & #1 > cat="NP"	--	#1:[cat="VP"] > [cat="PP"] & #1 > [cat="NP"]
Hierdurch werden alle VPn gefunden, die unmittelbar sowohl mindestens eine PP als auch mindestens eine NP enthalten. Die wörtliche Übersetzung der ANNIS-Suchanfrage lautet: »Ein Element ›VP‹ auf der Annotationsebene ›cat‹ dominiert ein Element ›PP‹ auf der Annotationsebene ›cat‹. Das erste Element (also ›VP‹) dominiert auch ein Element ›NP auf der Annotationsebene ›NP‹.« Die Paraphrasierung der TIGERSearch-Suchanfrage ist ganz ähnlich, nur dass sich die Bezeichnung der wiederaufgenommenen »VP« nicht aus der Reihenfolge der Elemente »VP« und »PP« ergibt, sondern dass das »VP«-Element explizit mit dem Namen »1« versehen wurde (jede andere Bezeichnung wäre hier möglich).		
cat="VP" > cat="PP" & #2 > cat="AP"	--	[cat="VP"] > #PP:[cat="PP"] & #PP > [cat="AP"]
Hierdurch werden alle VPn gefunden, die unmittelbar eine PP dominieren, die wiederum unmittelbar eine AP dominiert. Die TIGERSearch-Suchanfrage lautet paraphrasiert: »Ein Element mit dem Wert ›VP‹ auf der Annotationsebene »cat« dominiert unmittelbar ein als »PP« bezeichnetes Element ~~mit dem Wert »PP«~~ auf der Annotationsebene »cat«, und dieses Element »PP« dominiert unmittelbar ein Element mit dem Wert »AP« auf der Annotationsebene »cat«.		
cat="S" >* pos="KOUS" & #1 >* lemma="lieben"	--	#Satz:[cat="S"]>*[pos="KOUS"]& #Satz >* [lemma="lieben"]
Diese Suchanfragen finden Sätze, die sowohl eine Subjunktion (STTS: »KOUS«) als auch ein Token mit der Lemma-Annotation »*lieben*« enthalten. Potenziell werden also subjunktionale Nebensätze gefunden, die das Verb *lieben* beinhalten.		
cat="NP" >1,3 cat="S"	--	[cat="NP"] >1,3 [cat="S"]
Durch diese Suchanfragen wird der Grad der Abhängigkeit auf einen Mindest- und einen Höchstwert eingeschränkt: Es werden als »S« ausgewiesene Konstituenten (Sätze) gefunden, die entweder direkt abhängig oder bis zu einem Abhängigkeitsgrad von drei von einer als »NP« ausgewiesenen Konstituente ~~einer Nominalphrase~~ abhängt.		

Tab. 3.20: Beispielsuchanfragen für die Suche nach komplexen Dominanzgefügen zwischen Knoten

Arbeitsaufgaben

1. Die folgenden Suchaufgaben beziehen sich auf das TIGER-Korpus. Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2FpJDIU) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. ~~TüNDRA~~.
   a) Finden Sie alle Fälle, in denen eine NP unmittelbar das Lemma *Herz* dominiert.
   b) Finden Sie alle Fälle, in denen eine NP unmittelbar eine AP dominiert. (APn werden im TIGER-Korpus nur annotiert, wenn das Kopfadjektiv erweitert ist.)

c) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar ein pränominales Adjektiv und ~~(neben dem Kopfnomen, das braucht in der Suche nicht angegeben zu werden)~~ eine weitere NP enthält.

Hinweis: Formulieren Sie diese Anfrage nur für das ANNIS-Suchinterface, wenn Sie keinen Zugriff auf eine lokale Installation von TIGERSearch haben.

2. Die folgenden Suchaufgaben beziehen sich auf das TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL). Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2U1EZck) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. TüNDRA.
   a) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar eine Adjektivphrase dominiert. (Adjektivphrasen können im TüBa-D/Z-Korpus auch aus einer Worteinheit bestehen.)
   b) Finden Sie alle Fälle, in denen eine Adjektivphrase über genau zwei Generationen das Lemma *sehr* dominiert.

Beachten Sie, dass die Suche in ANNIS hypothetisch ist, da das Korpus nur in TüNDRA und als lokal in TIGERSearch zu installierende Datei verfügbar ist.

3.1.2.22 | Phrasenstrukturbaumbanken: Suche nach syntaktischen Funktionen

Wie in Kapitel 2.2.7.6 beschrieben wurde, besitzen manche Baumbanken zusätzlich zu Phrasenannotationen auch Annotationen von syntaktischen Funktionen. Auf diese Weise ist es nicht nur möglich, im syntaktischen Baum nach hierarchischen Beziehungen, also Mutter-Tochter- oder Schwester-Beziehungen zu suchen, sondern man kann den Typ der Beziehung spezifizieren und somit z. B. nach Objektbeziehungen und anderen syntaktischen Funktionen suchen. Technisch wird dies realisiert, indem einer Kante zwischen einem Mutter- und einem Tochterknoten eine Funktionsbezeichnung (ein Funktionslabel, z. B. »SB« für »Subjekt«) zugewiesen wird. Als Folge dessen kann man im Korpus nach Dominanzbeziehungen zwischen beliebigen Knoten (z. B. mit der Funktion »SB«) suchen und erhält als Treffer sämtliche Paare von Elementen, die diese Funktionsbeziehung haben. Siehe Tab. 3.21 für die Suche nach bestimmten Funktionsbeziehungen im TIGER-Korpus.

einfache Anf.

Arbeitsaufgaben

1. Die folgenden Suchaufgaben beziehen sich auf das TIGER-Korpus. Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2FpJDIU) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch ~~bzw. TüNDRA~~.
   a) Finden Sie alle Fälle, in denen eine VP unmittelbar eine NP als Akkusativobjekt dominiert.
   b) Finden Sie alle Fälle, in denen eine NP unmittelbar eine AP dominiert. ~~(APn werden im TIGER-Korpus nur annotiert, wenn das Kopfadjektiv erweitert ist.)~~

als Vergleichsphrase (Kantenbezeichnung: >CC*)

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
node >[func="OA"] node	--	#1 >OA #2
Die beiden Suchanfragen finden beliebige Elemente im Korpus, die in einer direkten Dominanzbeziehung zueinander stehen, die mit der Funktion »OA« (laut TIGER-Annotationsschema die Abkürzung für »Akkusativobjekt«) spezifiziert ist. ~~Man findet auf diese Weise sämtliche als Kopf-Akkusativobjekt-Gefüge ausgewiesene Syntagmen im Korpus!~~		
cat="NP" >[func="OP" pos="PROAV"	--	[cat="NP"] >OP [pos="PROAV"]
Im Gegensatz zum vorigen Beispiel sind in diesen Suchanfragen die Mutter und die Tochter der Dominanzbeziehung spezifiziert: Die Mutter ist eine NP (Nominalphrase), die Tochter hat den STTS-Wortartstatus »PROAV« für »Pronominaladverb«. Im TIGER-Korpus werden auf diese Weise Syntagmen gefunden wie *Eine Einigung darüber* (*habe es nicht gegeben*) (s47201 im 2012er Release).		
node >[func="OP"] cat="PP" > lemma="auf" (oder) node >[func="OP"] cat="PP" & #1 > lemma="auf"	--	#1 >OP #1:[cat="PP"] & #1 > [lemma="auf"]
Diese Suchanfragen finden Syntagmen mit einem Präpositionalobjekt, wobei der präpositionale Kopf »*auf*« selegiert wird. Da in der Suchanfrage die Mutter der Dominanzbeziehung nicht spezifiziert wird, werden sowohl verbale Köpfe (***warten auf***) als auch nominale Köpfe (***Reaktion auf***) und adjektivische (*eine auf dem Koran **beruhende** Ordnung*) gefunden.		

Tab. 3.21: Beispielsuchanfragen für die Suche nach Funktionsbeziehungen zwischen Knoten

c) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar ein pränominales Adjektiv und ~~(neben dem Kopfnomen, das braucht in der Suche nicht angegeben zu werden)~~ eine weitere NP enthält.

2. Das im ANNIS-Suchinterface durchsuchbare Korpus »pcc2.1« (das Potsdamer Zeitungskommentarkorpus; https://bit.ly/2Y9Rz8C) enthält ebenso Phrasenstrukturannotationen, welche nach dem TIGER-Korpus erstellt wurden.
Finden Sie in diesem Korpus alle Fälle, in denen eine VP eine PP als Präpositionalobjekt unmittelbar dominiert.

3. Die folgenden Suchaufgaben beziehen sich auf das TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL). Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2U1EZck) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. TüNDRA.
   a) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar eine Adjektivphrase dominiert. (Adjektivphrasen können im TüBa-D/Z-Korpus auch aus einer Worteinheit bestehen.)
   b) Finden Sie alle Fälle, in denen ein Akkusativobjekt im Vorfeld steht. (Die Suchanfrage muss so ausgedrückt werden, dass das Vorfeld eine beliebige Konstituente mit der Funktion Akkusativobjekt dominiert.)

Beachten Sie, dass die Suchanfragen für ANNIS wieder hypothetisch sind, weil TüBa-D/Z dort nicht verfügbar ist.)

3.1.2.23 | Phrasenstrukturbaumbanken: Weitere Suchfunktionen und -operatoren

Neben den bereits vorgestellten wesentlichen Suchfunktionen für die Suche in Phrasenstrukturbaumbanken gibt es einige weitere nützliche Funktionen, die gleichermaßen in den Suchprogrammen ANNIS und TIGERSearch bzw. TüNDRA ausgedrückt werden können. Diese sind in Tab. 3.22 zusammengefasst.

ANNIS	CQP/NoSketch Engine	TIGERSearch/TüNDRA
cat= "NP" >@l lemma= "schon"	--	[cat="NP"] >@l [lemma="schon"]
Der Operator »@l« berücksichtigt in direkten oder indirekten Dominanzen nur das ganz links stehende Kind einer Konstituente. Durch die angegebenen Anfragen werden somit Vorkommen von *schon* gefunden, sofern sie genau die ganz links stehende Tochter einer als »NP« ausgewiesenen Konstituente sind. Hiermit werden systematisch pränominale Vorkommen der Fokuspartikel *schon* gefunden.		
cat="NP">@r lemma="andererseits"	--	[cat="NP"] >@r [lemma="andererseits"]
Im Gegensatz zu dem Operator »@l« werden durch »@r« die ganz rechts stehenden Kinder einer Konstituente gefunden. Die abgebildeten Suchanfragen finden entsprechend dem Nomen (im Vorfeld) nachgestellten Adverb *andererseits*.		
cat="NP" & #1:tokenarity= 30,1000	--	#1:[cat="NP"] & tokenarity(#1,30,1000)
Diese Suchanfragen zielen darauf ab, als »NP« ausgewiesene Konstituenten zu finden, die mehr als 30 Token überspannen (die also verhältnismäßig groß sind). Da man bei diesen »Tokenarity«-Anfragen, wenn nicht eine genaue Anzahl, dann eine Spanne zwischen einer Mindest- und einer Höchstzahl spezifizieren muss, wird als Höchstwert eine relativ unrealistische Zahl angegeben. Bei dieser Suchanfrage spielt es keine Rolle, ob die von der Konstituente abhängigen Token direkt oder indirekt dominiert werden. Wenn ausschließlich direkt dominierte Elemente von der Zählung betroffen sein sollen, so verwenden Sie die folgenden Anfragen.		
cat="NP" & #1:arity= 8,1000	--	#1:[cat="NP"] & arity(#1,8,1000)
Diese Suchanfragen finden als »NP« ausgewiesene Konstituenten, die verhältnismäßig viele – und zwar mehr als acht – direkt abhängige Knoten besitzen. Während die vorige Anfrage einen Treffer erzeugt, wenn ein Nomen einen Relativsatz mit 29 Wörtern dominiert, sind die Anforderungen an die Struktur in diesem Fall von der Art, dass die »NP«-Konstituente neben dem Nomen und seinem Relativsatz mindestens sechs weitere direkte Dominanzen bzw. Dependenten benötigt.		

Tab. 3.22: Beispielsuchanfragen für die Suche in Phrasenstrukturbaumbanken mit bestimmten Suchoperatoren

3.1.2.24 | Dependenzstrukturbaumbanken: Suche nach syntaktischen Relationen

Da in Dependenzstrukturen ein Phrasenkonzept fehlt, kann man in Dependenzstrukturbaumbanken nur nach Beziehungen zwischen Wörtern (meistens also Token) suchen. Vergleichen Sie zu den prinzipiellen Suchmöglichkeiten, die Dependenzstrukturbaumbanken bieten, den zuvor be-

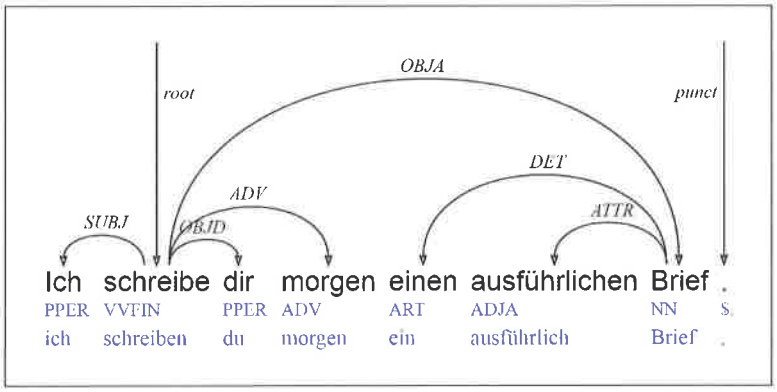


Abb. 3.10: Dependenzstrukturbaum zu dem Satz *Ich schreibe dir morgen einen ausführlichen Brief.* gemäß Foth (2006)

handelten Beispielsatz *Ich schreibe dir morgen einen ausführlichen Brief.* im Dependenz-Annotationsformat nach Foth (2006) (s. Abb. 3.10).

Die syntaktische Beschreibung des Beispielsatzes in Abb. 3.10 enthält Verbindungen zwischen den Wörtern bzw. Token im Satz und jeweils eine Funktion pro Beziehung. Diese Kanten-Annotationen können systematisch gesucht werden oder mit den vorhandenen Token-Annotationen (und weiteren Annotationen, falls vorhanden) in Beziehung gesetzt werden.

ANNIS	CQP/NoSketch Engine	TüNDRA
pos="ADJA" ->dep pos= "ADV"	--	[pos="ADJA"] > [pos="ADV"]
Die beiden Suchanfragen suchen nach pränominalen Adjektiven, die ein als »ADV« getaggtes Element (das STTS-Tag »ADV« beschreibt Adverbien und Partikeln) dominieren. Diese Suche findet typischerweise intensivierte pränominale Adjektive wie in *ein **sehr** schöner Morgen*.		
pos="ADJA" ->dep lemma= "sehr"		[pos="ADJA"] > [lemma= "sehr"]
Diese Suchausdrücke liefern die Intensivierungspartikel *sehr* im Zusammenhang mit pränominalen Adjektiven.		
pos="APPR" ->dep pos= "NN" & #1 . #2	--	#1:[pos="APPR"] > #2:[pos= "NN"] & #1 . #2
Diese Suchausdrücke finden unmittelbare Abfolgen von Präpositionen und Nomina, wobei die Präposition das Adverb unmittelbar dominiert (die untere Zeile der Suchanfragen drückt die unmittelbare Präzedenz aus). Hierdurch werden typischerweise Präpositionalphrasen wie *ab morgen* oder *für immer* gefunden, die keinen nominalen Kern besitzen.		
lemma="heute" ->dep lemma="noch" & #1 . #2	--	#1:[lemma="heute"] > #2:[lemma="noch"] & #1 . #2
Diese Suchanfragen finden die Abfolge *heute noch* (untere Zeile der Suche). Durch die angegebene Dominanzbeziehung (*heute* dominiert *noch*) wird sichergestellt, dass die beiden Elemente nicht zufällig nebeneinander vorkommen (wie in *Ich habe heute noch mehr Appetit als gestern*).		

Tab. 3.23: Beispielsuchanfragen für die Suche nach Dependenzbeziehungen zwischen Token. Diese Suchanfragen erzielen für das Korpus »tiger_dep_v2.2« im ANNIS-Interface sowie für das Korpus »TüBa-D/Z 10 Dependency (Experimental)« im TüNDRA-Suchinterface Treffer.

Dependenzen sind immer gerichtet, deshalb muss hier genauso wie bei der Suche in Phrasenstrukturen eine Mutter und eine Tochter spezifiziert werden. In der Regel ist die Gerichtetheit der Dependenzen von der Mutter zur Tochter, also nach unten orientiert. So wird die Suche nach einem Verb mit einem Tochter-Nomen syntaktische Beziehungen des Typs Verb-Subjekt, Verb-Argument oder Verb-Adjunkt ausgeben. Bei der Suche nach Nomen-Artikel-Beziehungen werden sämtliche Nomina mit Artikel ausgeben und Nomina ohne Artikel unberücksichtigt gelassen. Siehe Tab. 3.23 für weitere Beispiele sowie konkrete Suchanfragen. Beachten Sie, dass von den ausgewählten Suchprogrammen nur ANNIS und TüNDRA Dependenzstrukturbaumbanken verarbeiten können.

Hinweis: Beachten Sie, dass der Operator für Dependenzbeziehungen in ANNIS variieren kann. Anstelle des oben verwendeten Operators »->dep« wird in bestimmten Korpora auch »->dependency« oder eine schließende Spitzklammer »>« verwendet. Mittels des Informationsknopfs »i« neben den entsprechenden Korpora und dem Anwählen von »Edge Annotations« gelangt man zu Beispielsuchanfragen, die einem Aufschluss über jeweilige Bezeichnung geben.

Arbeitsaufgabe

Die folgenden Suchaufgaben beziehen sich auf das Korpus »tiger_dep_v2.2« im ANNIS-Suchinterface sowie das Korpus »TüBa-D/Z 10 Dep[illegible]« im TüNDRA-Suchinterface.

a) Finden Sie alle Vorkommen des Lemmas *schön*, die unmittelbar von einem Nomen dominiert werden.

b) Finden Sie Adverbien, die unmittelbar von Verben dominiert werden.

3.1.2.25 | Dependenzstrukturbaumbanken: Suche nach syntaktischen Relationen mit Funktionen

Noch komplexer als in den bisher behandelten Suchen werden die Suchanfragen, wenn man die Suche zusätzlich auf syntaktische Funktionen wie bestimmte Objekttypen einschränken möchte. Siehe Tab. 3.24 für entsprechende Beispiele.

Hinweis: Beachten Sie, dass nicht nur der Operator für Dependenzbeziehungen in ANNIS, sondern auch der Variablenname für die Kantenbezeichnung variieren kann. Anstelle des oben verwendeten Variablennamens »deprel« wird in bestimmten Korpora auch »func« verwendet. Mittels des Informationsknopfs »i« neben den entsprechenden Korpora und dem Anwählen von »Edge Types« gelangt man zu Beispielsuchanfragen, die einem Aufschluss über jeweilige Bezeichnung geben.

ANNIS	CQP/NoSketch Engine	TüNDRA
node ->dep[deprel="OP"] node	--	[word=/.*/] >OBJP [word=/.*/]
Durch diese Anfragen werden sämtliche Funktionsbeziehungen im Korpus gefunden, die mit dem Wert »OP« für »Präpositionalobjekt« versehen sind.		
pos=/V.*/ ->dep[deprel="OP"] pos="PROAV"	--	[pos=/V.*/] >OBJP [pos="PROP"]
Im Gegensatz zum vorigen Beispiel sind in diesen Suchanfragen die Mutter und die Tochter der Dominanzbeziehung spezifiziert: Die Mutter ist ein Verb, die Tochter hat den STTS-Wortartstatus »PROAV« für »Pronominaladverb« (bzw. das entsprechende Tag »PROP« im TüBa-D/Z-Korpus). In den Korpora werden auf diese Weise Syntagmen gefunden wie *Eine Einigung darüber* (*habe es nicht gegeben*).		
node ->dep[deprel= "OP"] pos="APPR" & #2 _=_ lemma="auf"	--	[word=/.*/] >OBJP [pos="APPR" & lemma="auf"]
Diese Suchanfragen finden Syntagmen mit einem Präpositionalobjekt, wobei der präpositionale Kopf »*auf*« selegiert wird, was durch den Lemma-Zusatz spezifiziert wird. Da in der Suchanfrage die Mutter der Dominanzbeziehung nicht spezifiziert wird, werden sowohl verbale Köpfe (***warten** auf*) als auch nominale Köpfe (***Reaktion** auf*) und adjektivische (*eine auf dem Koran **beruhende** Ordnung*) gefunden.		

Tab. 3.24: Beispielsuchanfragen für die Suche nach Dependenzbeziehungen zwischen Token mit Funktionsausweisungen. Diese Suchanfragen erzielen für das Korpus »tiger_dep_v2.2« im ANNIS-Interface sowie für das Korpus »TüBa-D/Z 10 Dependency (Experimental)« im TüNDRA-Suchinterface Treffer.

Arbeitsaufgabe

Die folgenden Suchaufgaben beziehen sich auf das Korpus »tiger_dep_v2.2« im ANNIS-Suchinterface sowie das Korpus »TüBa-D/Z 10 Dependency (Experimental)« im TüNDRA-Suchinterface.

a) Finden Sie alle Nomina, die andere Elemente als Akkusativobjekte unmittelbar dominieren. (Das Funktionslabel für Akkusativobjekte im TIGER-Korpus ist »OA«, im TüBa-D/Z-Dep-10-Korpus »OBJA«.)

b) Finden Sie alle Präpositionen, die unmittelbar von Nomina als (postnominale) Modifikatoren abhängen. (Das Funktionslabel für postnominale Modifikatoren im TIGER-Korpus ist »MNR«, im TüBa-D/Z-Dep-10-Korpus »PP«.)

3.1.2.26 | Diskurslinguistische Kategorien: Suche nach Koreferenzen mittels ‚pointing relations'

Im Kontext diskurlinguistischer Analysen werden Annotationstypen verwendet, die mit ‚pointing relations' oder ‚reference relations' bezeichnet werden (s. auch Kap. 2.2.8). Diese Konzepte stehen für Referenzbeziehungen im Text, wie z. B. der Anaphorik eines Personalpronomens hinsichtlich eines vorher eingeführten Diskursreferenten. Durch die Annotation werden Token oder komplexere Ausdrücke, die mehrere Token um-

Tab. 3.25: Beispielsuchanfragen für die Suche nach Koreferenzbeziehungen (die Suchanfragen gelten für das Korpus »pcc2« im ANNIS-Suchinterface der Humboldt-Universität zu Berlin)

ANNIS	CQP/NoSketch Engine	TüNDRA
node ->anaphor_antecedent node	--	--
Durch diese Anfrage werden im PCC-Korpus sämtliche Fälle von koreferenten Ausdrücken gefunden. Hierbei spezifiziert der Pfeil-Operator das Annotationskonzept (pointing relation bzw. Referenzbeziehung) und das Label »anaphor_antecedent« die Art der Referenzbeziehung. Jeder Treffer enthält genau einen koreferenten Ausdruck und den in der Koreferenzkette nächstliegenden zugehörigen Ausdruck. Um die beteiligten koreferenten Ausdrücken weiter zu spezifizieren, kann die Suchanfrage wie folgt erweitert werden.		
node ->anaphor_antecedent node & #1 _i_ pos="NN"	--	--
Im Gegensatz zum vorigen Beispiel werden bei dieser Anfrage nur koreferente, wiederaufnehmende Elemente gefunden, die ein Nomen enthalten (in der Regel könnten die wiederaufnehmenden Ausdrücke pronominal sein, was hier ausgeschlossen wird). Dies wird dadurch erreicht, dass der Knoten, der auf einen früheren koreferenten Knoten verweist, ein Element mit der Wortartannotation »NN« (nach STTS »normales Nomen«) enthalten soll. Alternativ können statt »NN« »PPER« oder andere Wortarten wie Pronominaladverbien (STTS: PROAV), die koreferent sein können, eingesetzt werden.		

fassen, miteinander verknüpft, ähnlich wie bei der Annotation von Dependenzbeziehungen. Gegenüber den Dependenzen hat die Annotation mit ‚pointing relations' jedoch nicht die Bedeutung der Abhängigkeit und damit eine klar hierarchische Lesart (die Verbindung zweier Dependenten in der Dependenzannotation markiert nämlich immer einen Teil als den übergeordneten und einen als den untergeordneten). ‚Pointing relations' drücken aus, dass zwei oder mehr Elemente identisch sind bzw. auf dieselbe Entität verweisen. Somit sollten Dependenzen und ‚pointing relations' als Annotationstypen für verschiedene linguistische Konzepte unterschieden werden.

Ein Korpus mit Koreferenzannotationen mittels ‚pointing relations' ist das »Potsdam Commentary Corpus« (PCC; http://angcl.ling.uni-potsdam.de/resources/pcc.html), das aus Zeitungskommentartexten besteht und eines der am reichhaltigsten annotierten Korpora ist. Es ist frei verfügbar zum Download und kann im ANNIS-Interface der Humboldt-Universität zu Berlin (https://korpling.german.hu-berlin.de/annis/) In einem Ausschnitt von zwei Texten getestet werden (der Korpusname ist »pcc2«). Die Suchanfragen in Tab. 3.25 beziehen sich auf diese Korpusdaten.

Arbeitsaufgabe

Testen Sie die angegebenen Suchanfragen mit den genannten wortartmäßigen Varianten zu verschiedenen Koreferenzausdrücken im PCC-Korpus, das Sie über das ANNIS-Suchinterface abfragen können. Überlegen Sie, welche Wortarten zusätzlich zu den genannten koreferent sein können, und testen Sie die Annahmen am genannten Korpus.

3.1.2.27 | Suche unter Einbezug von Metadaten (Berücksichtigung von Kontextfaktoren)

Für Metadaten gilt wie für die Suche nach Annotationen im Allgemeinen, dass im Korpus Variablen und Werte vergeben wurden, die in der Suche berücksichtigt werden können. Bei einem Suchausdruck mit Metadaten-Zusatz handelt sich gewissermaßen um einen Filtervorgang, bei dem man bestimmte Teile eines Korpus anhand bestimmter Variablen (Texttypen, Kontextfaktoren, Typen von Sprecherinnen und Sprechern usw.) gezielt ausschließen oder isolieren kann. Siehe Tab. 3.26 für Beispiele.

~~Beachten Sie die Arbeitsaufgabe 1 für eine entsprechende Übung und Suchen in bestimmten Korpora!~~

Im Folgenden werden authentische Korpussuchen mit Metadatenspezifikationen angegeben. Bei der Speicherung von Metadaten existieren kaum Standards und viele Korpora werden ganz ohne (annotierte) Metadaten angeboten.

Die vorgestellten Korpora mit bestimmten Metadaten dienen als individuelle Beispiele. Andere Korpora besitzen je nach Zusammensetzung ganz andere Metadaten.

Nutzen Sie parallel zu den folgenden Ausführungen die jeweils erwähnten Korpora des CQP-Webinterfaces sowie des ANNIS-Suchinterfaces.

Szenario 1 zum Korpus »Parlamentsreden« im CQP-Webinterface der Humboldt-Universität zu Berlin: Das Korpus enthält protokollierte Bundestagsdebatten (Plenarprotokolle), die auf der Webseite des Deutschen

ANNIS	CQP/NoSketch Engine
SUCHANFRAGE & meta::VARIABLE="WERT"	SUCHANFRAGE ::match.VARIABLE=- "WERT"
In beiden Fällen hängt man an eine Suchanfrage also eine Metadatenrestriktion an. Dies setzt voraus, dass das gegebene Korpus Metadatenannotationen enthält.	
pos="PTKVZ" & meta::Genre="Roman"	[pos="PTKVZ"] ::match.Genre="Roman"
Für diese Anfragen wurde angenommen, dass ein Korpus existiert, das verschiedene Genres enthält, dass diese unter dem Variablennamen »Genre« gekennzeichnet sind und dass ein im Korpus enthaltener Wert »Roman« existiert. Die eigentliche Suchanfrage findet laut dem STTS als Verbpartikeln (»PTKVZ«) ausgewiesene Wortformen. Es würden nur Vorkommen gefunden werden, die in dem Teil des Korpus mit dem spezifizierten Metadatum enthalten sind.	

Tab. 3.26: Allgemeines Schema und ~~kon-krete~~ Beispiel zur Suche mit Metadatenrestriktionen in den Suchsystemen ANNIS und CQP bzw. NoSketch Engine

Bundestags frei zur Verfügung gestellt werden (http://www.bundestag.de/protokolle). Die im Korpus zusammengestellten Daten reichen von 1996 bis 2002. Stellen wir uns vor, man ist daran interessiert, die zu dieser Zeit aufkommende Kommunikationsform E-Mail in den Debatten zu untersuchen. Dann kann man das Metadatum der Jahreszahl nutzen, um die einzelnen Erhebungsjahre getrennt abzufragen. Eine entsprechende Suchanfrage ist

```
[word="e-?mails?" %c] ::match.quelle_year="1996"
```

Der Suchausdruck in eckigen Klammern lässt verschiedene Schreibungen des Begriffs ‚E-Mail' zu (Groß- und Kleinschreibung wird nicht unterschieden, der Bindestrich ist optional und ein Plural-*s* ist auch optional). Durch die Erweiterung dieses Suchausdrucks werden nur die Protokolle des Jahres 1996 durchsucht. Die Folgejahre bis 2002 können in separaten Anfragen erfasst und die Suchergebnisse vergleichend ausgewertet werden.

Beachten Sie die Arbeitsaufgabe 2 zu diesem Nutzungsszenario.

Szenario 2 zum Korpus »Akademisches Deutsch« im CQP-Webinterface der Humboldt-Universität zu Berlin: In diesem sind Dissertationsabstrakts zu verschiedenen Fachgebieten gesammelt und korpuslinguistisch aufbereitet worden, so dass mithilfe dieses Korpus Studien zur Wissenschaftssprache, auch vergleichend über verschiedene Fachgebiete durchgeführt werden können. Ein Untersuchungsszenario könnte sein, über die vergleichende Auswertung gewisser Inhaltswörter die fächerspezifischen Interessensgebiete und Forschungsstrategien zu analysieren. Eine entsprechende Suchanfrage nach Nomina, die lediglich Treffer im Fachbereich Chemie liefert, ist:

```
[pos="KOUS"] ::match.abstract_sachgebiet="chemie"
```

Verwenden Sie bei der Trefferausgabe in CQP die Option »frequencies«, um nicht Treffer für Treffer zu sehen, sondern eine Frequenzliste zu erzeugen. Andere im Korpus enthaltene Fachbereiche sind »psychologie«, »politik« und »paedagogik«. ~~Mittels des Suchinterfaces kann zunächst eine Auflistung aller verarbeiteten Fachgebiete erstellt werden, um zu wissen, welche Metadatenwerte verfügbar sind.~~

Hinweis: Zu quantitativen Vergleichen (z. B. zum Vergleich der jeweiligen Häufigkeit bestimmter Wortarten in den Subkorpora des Korpus) müssen Sie normalisieren. Diese Operation wird in Kapitel 4.5.1 ~~behandelt~~. Arbeitsaufgabe 3 am Ende dieses Kapitels behandelt den quantitativen Vergleich von Subjunktionen in drei Subkorpora des Korpus »Akademisches Deutsch«.

~~Beachten Sie die Arbeitsaufgabe 3 zu diesem Nutzungsszenario!~~

Szenario 3 zum Korpus »Falko« im ANNIS-Webinterface der Humboldt-Universität zu Berlin: Falko (http://hu-berlin.de/falko) ist ein Lernerkorpus mit fortgeschrittenen Lernenden des Deutschen als Fremdsprache mit unterschiedlicher sprachlicher Herkunft bzw. heterogenem Muttersprachenhintergrund. Das Korpus ist u. a. in einer eigenen ANNIS-

Instanz verfügbar (http://hu-berlin.de/falko-suche) und kann darüber abgefragt werden. Ein häufig angewandtes Szenario dabei ist die getrennte Abfrage nach bestimmten Herkunftssprachen (L1). ~~Fragt man sich~~ z. B., ob bestimmte Modalpartikeln von den Lernenden bestimmter L1-Kontexte überhaupt verwendet werden und ob es Häufigkeitstendenzen gibt. Eine entsprechende Suchanfrage für das Korpus »falkoEssayL2v2.4« ist:

```
word=/(wohl|eh|ja|halt)/ & meta::l1_1="eng"
```

Diese Suchanfrage findet die aufgeführten Formen *wohl*, *eh*, *ja* und *halt* (natürlich können hierdurch jeweils falsche Treffer gefunden werden, die entsprechend herausgefiltert werden müssten). Der Metadatenzusatz berücksichtigt ausschließlich Sprecherinnen und Sprecher der Muttersprache Englisch. Weitere Länderkürzel sind »dan« für Dänisch, »rus« für Russisch, »fra« für Französisch (weitere Ländercodes können dem Falko-Handbuch entnommen werden: http://hu.berlin/falko_handbuch).

Arbeitsaufgaben

1. In der Korpussuchmaschine ANNIS finden Sie das Korpus »Maerchenkorpus«, bestehend aus 211 Märchen und Kinderlegenden der Gebrüder Grimm. Sie sind mit dem üblichen automatischen Verfahren getaggt, so dass sie nach Lemmata und STTS-Wortarten durchsuchbar sind. Die einzelnen Märchen sind in der Suche mit der Metadaten-Variable »Titel« voneinander trennbar. Zwei der Märchentitel sind »Sneewittchen« und »Daumesdick«.
   a) Finden Sie alle Fälle des Lemmas *Apfel* in dem Märchen »Sneewittchen«.
   b) Führen Sie dieselbe Suche ohne Metadateneinschränkung durch und schauen Sie, ob das Nomen überhaupt außerhalb des Schneewittchen-Märchens in der Märchensammlung auftritt.
   c) Finden Sie alle Wörter in dem Märchen »Daumesdick«, die gemäß dem STTS-Tagset als Adverbien ausgewiesen sind.

2. (Zu Szenario 1): Wenden Sie die eingeführte Suche auf die einzelnen Jahre von 1996 bis 2002 an und schauen Sie nach offensichtlichen Veränderungen in der Verwendung über die Jahre hinweg.

3. (Zu Szenario 2): Vergleichen Sie die Vorkommen Nomina in den Fachgebieten Chemie (»chemie«), Physik (»physik«) und Medizin (»medizin«) und schauen Sie, ob Sie qualitative Unterschiede ausmachen können.

4. (Zu Szenario 3):
   a) Vergleichen Sie die Vorkommen von Modalpartikeln in den genannten Falko-Lernergruppen L1 Englisch, Dänisch, Russisch und Französisch im Korpus »falkoEssayL2v2.4« des ANNIS-Suchinterfaces.
   b) Schauen Sie sich anschließend die Treffer im Korpus »falkoEssayL1v2.3« an. Sie müssen hier den Metadatenzusatz weglassen, weil das Vergleichskorpus ausschließlich aus deutschsprachigen Muttersprachlerinnen und Muttersprachlern besteht und deshalb für die Variable »L1« bzw. »Muttersprache« keine Metadatenannotation benötigt wird.

   Hinweis: Auch an dieser Stelle dürfen Sie nur qualitativ und nicht quantitativ vergleichen, weil Sie Datenmenge der jeweiligen Lernerkohorte nicht kennen und deshalb nicht einschätzen können, ob die Anzahl der gefundenen Treffer gemessen an der jeweiligen Gesamtdatenmenge hoch oder niedrig ist. Vergleichen Sie zum Konzept des Normalisierens Kap. 4.5.1.

3.1.2.28 | ANNIS: Exportmöglichkeiten und Quantifizierbarkeit der Ergebnisse

ANNIS bietet zu einer gegebenen Suchanfrage verschiedene Exportmöglichkeiten in dem Format CSV (tabellarisch separierte Werte) und im Textformat an. Derzeit stehen sieben Exporter mit verschiedenen Funktionen zur Auswahl, die in der folgenden Auflistung zusammengefasst sind:

- »**CSVExporter**« und »**CSVMultiTokExporter**«: Die der Suche entsprechenden Elemente werden Token-Treffer mitsamt Annotationen und Metadaten (diese müssen spezifiziert werden) herausgeschrieben. Der Trefferkontext wird nicht exportiert. Exportiert wird eine Textdatei, in der verschiedene Werte pro Treffer durch Tabulatorabstände getrennt sind.
- »**WekaExporter**«: Dieser Exporter exportiert ebenso Token-Treffer ohne Kontext mit von den Nutzern ausgewählten Annotationen und Metadaten. Die Abgrenzung der exportierten Werte erfolgt mit Kommata. Die Werte befinden sich in einfachen Anführungszeichen. Das Format der Datei ist passend für das statistische Auswertungsprogramm Weka (vgl. Witten/Frank/Hall 2011, S. 403 f., http://www.cs.waikato.ac.nz/~ml/weka/index.html).
- »**GridExporter**«: Die gefundenen Elemente werden mit einem spezifizierten Kontext untereinander in eine Textdatei geschrieben, wobei eine Korpusebene (Token oder eine beliebige Annotationsebene) exportiert wird. Metadaten werden dabei berücksichtigt.
- »**SimpleTextExporter**«: Zu einer Suchanfrage wird lediglich der abgedeckte Text auf der Tokenebene exportiert. Zu jedem Treffer können Metadaten exportiert werden.
- »**TokenExporter**«: Wie der vorige Exporter; hier werden sämtliche Token-Annotationen (durch Schrägstriche separiert) mit den Token exportiert.
- »**TextColumnExporter**«: Es wird eine mehrspaltige Tabelle erzeugt, in der Metadaten zu jedem Treffer, der linke sowie der rechte Trefferkontext in verschiedenen Spalten stehen.

Zur Anwendung dieser Exportfunktionen s. Kap. 4.3 sowie Kap. 4.6.1.

Im Folgenden werden die nötigen Verarbeitungsschritte gemäß dem Anwendungsszenario, dass sämtliche Eigennamen aus dem im ANNIS-Suchinterface (https://hu.berlin/annis-intro) verfügbaren Korpus »Fuerstinnenkorrespondenz1.1« herausgeschrieben werden sollen, beschrieben.

Anleitung

- Wählen Sie ein Korpus aus der Korpusliste links unten im Interface.
  Konkretes Beispiel: Fuerstinnenkorrespondenz1.1
- Geben Sie oben rechts eine gültige Suchanfrage ein.
  Konkretes Beispiel: Suchanfrage: pos = "NE"
- Klicken Sie auf den Menüknopf »More« und wählen Sie »Export«.
- Wählen Sie einen passenden Exporter und stellen Sie jeweils den Ausgabekontext auf 0.
  Konkretes Beispiel: TokenExporter
- Betätigen Sie »Perform Export« und warten Sie, bis der Export fertiggestellt ist.
- Betätigen Sie »Download« und speichern Sie die ausgegebene Datei lokal auf Ihrem Computer.
  Sie erhalten eine Liste der Treffer ohne jeglichen Satzkontext. Wie man mit einer solchen Liste weiterarbeiten kann, wird in Kap. 4.3 behandelt.

Zum Export von Token-Treffern mit Trefferkontext durchlaufen Sie die Schritte, wie in der Anleitung beschrieben, stellen jedoch den Ausgabekontext auf einen der möglichen Werte. Sie erhalten eine Ausgabe mit dem Satzkontext des jeweiligen Treffers.

Zur statistischen Auswertung bietet ANNIS die Möglichkeit, Frequenzlisten zu erstellen, als Säulendiagramme darzustellen und die Frequenzdaten als CSV-Tabelle herunterzuladen (s. z. B. Kap. 4.3).

3.1.2.29 | CQP und NoSketch Engine: Exportmöglichkeiten und Quantifizierbarkeit der Ergebnisse

Die Exportmöglichkeiten von CQP und NoSketch Engine sind äußerst vielseitig. Zu einer Suchanfrage mit einem gesuchten Element (z. B. einem Verb: [pos = "V.*"]) lässt sich für die Trefferausgabe eine prinzipiell beliebige Kontextgröße festlegen (der Zusatz »set leftContext 3 word« in der CQP-Anfragesprache spezifiziert z. B. den linken Ausgabekontext auf drei Wörter). Es lassen sich auch beliebige im Korpus verfügbare Annotationen herausschreiben, ebenso Metadaten bzw. strukturelle Attribute. Es lassen sich zu allen verfügbaren Werten Frequenzlisten erstellen (man kann sich z. B. zu der oben gegebenen Suchanfrage »[pos = "V.*"]« eine Liste mit Lemmata oder Wortformen ausgeben lassen, sortiert nach der Häufigkeit der jeweiligen Lemmata bzw. Wortformen.

Bei Anfragen mit mehreren Elementen (z. B. Abfolgen von Adjektiven und Nomina) lassen sich solche Listen nach Frequenzen für eines der gesuchten Elemente oder der Kombination der Elemente erstellen. Die

genannten CQP- und NoSketch-Engine-basierten Suchinterfaces bieten allerdings, was die Suchausgabe angeht, jeweils beschränkte und sehr unterschiedliche Funktionen an. Deshalb beziehen sich die folgenden Erläuterungen auf eine Beispielressource, das CQP-Webinterface der Humboldt-Universität zu Berlin (Internetadresse: https://hu.berlin/cqp; Nutzername: CQP_Demo; Passwort: TestSuchen).

Szenario zum Export von Wortbildungsprodukten, die auf *-weise* enden (im Korpus »Parlamentsreden«): Um ein solches Anliegen in einer Korpussuche umzusetzen, kann man nach Wortformen suchen, die auf *-weise* enden; gleichzeitig kann man die Wortarten Verb und Nomen ausschließen, wenn das Verb *weisen* und das Nomen *Weise* in dem Studienkontext als irrelevant erachtet werden. Die Suchanfrage in CQP bzw. für NoSketch Engine lautet dann:

```
[word=".+weise" & pos!="(NN|VV.*)"]
```

Um eine Liste von Suchergebnissen zu generieren, können Sie im genannten Interface wie folgt vorgehen.

Anleitung

- Loggen Sie sich mit den oben genannten Daten ein.
- Wählen Sie oben rechts im Interface das Korpus »Parlamentsreden« aus.
- Geben Sie oben links die genannte Suchanfrage ein.
- Stellen Sie den linken und rechten Ausgabekontext auf eine geeignete Größe, z. B. zehn Stellen links und rechts.
- Klicken Sie »Search«, um die Suche abzuschicken.
- Kopieren Sie den Inhalt aus dem sich öffnenden Fenster in eine Textdatei, indem Sie einen beliebigen Texteditor verwenden. Bei Editoren wie dem LibreOffice (oder OpenOffice) Writer oder Microsoft Word, die Textformatierungen zulassen, werden die Tabellendarstellung und die Fettmarkierung des eigentlichen Treffers übernommen.

Um lediglich eine Liste von Wortformen zu erhalten, gehen Sie vor, wie oben beschrieben, und berücksichtigen die folgenden Änderungen.

Anleitung

- Stellen Sie den Ausgabekontext möglichst gering ein (eine Stelle links und rechts).
- Stellen Sie das Ausgabeformat (»output format«) auf »Plain«, so dass unformatierter Text ausgegeben wird, in welchem der eigentliche Treffer jeweils durch Spitzklammern markiert ist.
- Nun muss man die Treffer isolieren, indem man den linken und den rechten Kontext entfernt. Wie man dies tut, wird in Kap. 4.1 besprochen.

Um gleich in der CQP-Ausgabe eine Frequenzliste zu erhalten, gehen Sie vor, wie oben beschrieben, und berücksichtigen die folgenden Änderungen.

Anleitung

- Stellen Sie die Ausgabe (»Output«) auf »frequencies«. Nun werden im Ausgabefenster Frequenzen der einzelnen Formen der Häufigkeit nach geordnet aufgelistet. Groß- und kleingeschriebene Formen werden dabei separat gezählt.
- Wenn man die in der Ausgabe berücksichtigte Größe (jede der unter »positional attributes« verfügbaren Größe kann angewählt werden) ändert, wird die Zählung entsprechend hinsichtlich der ausgewählten Variable durchgeführt. Im gewählten Korpus lassen sich somit auch Lemmata und Wortarten quantifizieren.

3.1.2.30 | TIGERSearch: Exportmöglichkeiten und Quantifizierbarkeit der Ergebnisse

Die im Ergebnisfenster angezeigten Daten lassen sich ausschließlich Graph für Graph betrachten. Möchte man sämtliche Treffer in eine Datei exportieren und somit zusammenführen, so wählt man im Hauptfenster (nicht im Ergebnisausgabefenster) unter dem Menüpunkt »Query« > »Export matches«. Gehen Sie in dem sich öffnenden Fenster »Export Options« wie folgt vor.

Anleitung

- Wählen Sie als Exportformat (in der Box »Export format«) »XML piped through XSLT«.
- Klicken Sie »Search«, wählen ein Verzeichnis und geben Sie anschließend einen Dateinamen (die Datei muss noch nicht existieren) inklusive Dateiendung (also z. B. »Export.txt«) ein. Klicken Sie auf »Select«.
- Klicken Sie auf »Submit«. Es wird eine Textdatei erzeugt, die sämtliche der Suche entsprechenden Treffersätze enthält, wobei standardmäßig die konkreten Treffer im Satz nicht markiert werden.
  - Um dies zu erreichen, wählen Sie rechts unten in der Box »XML piped through XSLT« die Einstellung »sentence format (all tokens, matching tokens marked)«.
  - Um lediglich Treffer ohne ihren Satzkontext herauszuschreiben, wählen Sie statt »sentence format (all tokens)« »sentence format (all matching tokens)«.
  - Wie Sie mit solchen Trefferlisten weiterarbeiten und die Treffer quantifizieren können, wird in Kap. 4.1 bis Kap. 4.3 besprochen.

Will man Treffer gleich in TIGERSearch quantifizieren, so hilft einem die Funktion »Query« > »Statistics« im Hauptfenster (nicht im Suchausgabefenster). Hierbei ist es wichtig, dass man zuvor in dem eingegebenen Suchausdruck Variablennamen für die auszuwertenden Elemente vergeben hat.

Szenario zur Frequenzauswertung in TIGERSearch: Stellen Sie sich vor, Sie wollen unmittelbare Abfolgen von Präpositionen und Nomina untersuchen. Der entsprechende Suchausdruck in TIGERSearch lautet:

```
[pos="APPR"] . [pos="NN"]
```

Mit Variablennamen sieht dieser Ausdruck z. B. wie folgt aus:

```
#Präposition:[pos="APPR"] . #Nomen:[pos="NN"]
```

Es genügen Namen aus einem Buchstaben oder einer Ziffer, z. B. »P« für »Präposition« und »N« für »Nomen«.

Um nun die angegebenen Variablen frequenzmäßig auszuwerten, gehen Sie wie folgt vor.

Anleitung

- Öffnen Sie das Auswertungsfenster durch Klicken auf »Query« und »Statistics«.
- Klicken Sie auf das obere freie Auswahlkästchen unter »Feature 1«.
- Wählen Sie das auszuwertende Element aus.
- Wählen Sie im darunterliegenden Auswahlkästchen eine der für das Element verfügbaren Variablen aus. Wenn Sie nach Lemmata quantifizieren wollen, wählen Sie also »lemma«.
- Klicken Sie auf den Menüpunkt »Frequency«.
- Klicken Sie auf den Menüpunkt »Build«.
- Sie erhalten eine Liste mit nach Frequenzen sortierten Werten, die Sie weiterhin mit der Funktion »Export« exportieren können.
  - Um eine Textdatei zu erzeugen, wählen Sie hier im Exportfenster das Format »Text«, klicken Sie auf »Search«, spezifizieren Sie einen Ausgabeordner und geben Sie einen Dateinamen samt Endung (z. B. »Frequenzen.txt«) an.
  - Klicken Sie auf »Speichern« im zweiten Fenster und auf »Export« im ersten. Es wird eine Datei mit den Daten des Statistikfensters erzeugt. Es lassen sich auch Microsoft-Excel-kompatible Dateien und XML-Dateien erzeugen.

3.2 | Online-Suchinterfaces für große Standardkorpora

In den vorangegangenen Kapiteln zur Korpussuche wurden detaillierte Korpussuchfunktionen vorgestellt, die sämtliche Annotationstypen abdecken, die im Oberkapitel zur Korpusaufbereitung (s. Kap. 2) behandelt wurden. In den kommenden Kapiteln hinsichtlich der großen Online-Plattformen zur Korpusrecherche in deutschen Korpora wird gezeigt, ob und wie diese Suchfunktionen verfügbar sind. Dafür wird ein Set von Anforderungen formuliert und mit den Möglichkeiten der einzelnen

~~Suchsysteme abgeglichen. Die folgende Auflistung bezeichnet die einzelnen Anforderungen.~~ Dabei werden die folgenden Informationen zur Handhabung des Systems zusammengetragen:

- Auswahl von Korpusdaten,
- Suche nach Wortformen (*case sensitive* und *case insensitive*),
- Suche nach Lemmata,
- Suche nach Wortarten,
- Suche mit regulären Ausdrücken,
- Suche nach Abfolgen sowie
- Suche nach mehreren Eigenschaften desselben Token.

~~Sofern das entsprechende Suchsystem besondere Merkmale aufweist, werden weitere Suchfunktionen angegeben.~~

Auf die Prüfung dieser Suchfunktionalitäten folgen abschließend die Aspekte

- Quantifizierbarkeit der Ergebnisse sowie
- Exportmöglichkeiten.

3.2.1 | DWDS und DTA

Das DWDS (»Digitales Wörterbuch der deutschen Sprache«, ~~http://~~www.dwds.de) ist eine korpusbasierte Onlineplattform, die von den hier behandelten online-Ressourcen die meisten Auswertungsfunktionen besitzt und am leichtesten zugänglich ist. Letzteres liegt daran, dass die Nutzung des DWDS (zumindest in seiner Kernfunktion) keinerlei Registrierung erfordert. Dasselbe gilt für das DTA (»Deutsches Textarchiv«, ~~http://~~www.deutschestextarchiv.de/), das zum einen eine historische Komponente des DWDS ist und zum anderen eine eigene Internetpräsenz mit vergleichbaren Korpussuchfunktionen besitzt.

Grundlegend ist das DWDS als Lexikon und als Korpusressource mit einer Wort-im-Kontext-Ausgabe sowie mit verschiedenen statistischen Instrumenten nutzbar. Wir werden uns auf letzten beiden Funktionen ~~beschränken~~, werden aber die Lexikonfunktion kurz ~~als erstes~~ beleuchten.

RF Konzentrieren

Allgemeine Informationen und das DWDS-Lexikon: Besucht man die Homepage des DWDS (~~http://~~www.dwds.de), so kann man unmittelbar Suchbegriffe in das obige Textfenster eingeben. Tippt man einzelne Wörter ein, so bietet das System automatisch verschiedene Funktionen an, allen voran einen Wörterbucheintrag, sofern im System existent. Dieser enthält Informationen zur Grammatik des Worts (Wortart, Informationen zur Flexion, zu alternativen Formen, zur Silbentrennung und morphologischen Zusammensetzung), zur Bedeutung (mit Wortdefinitionen, getrennt nach verschiedenen Lesarten bei Polysemen und Homonymen), zur Etymologie (nach ~~Wolfgang~~ Pfeifer ~~ohne Datum angegeben~~), außerdem wird ein Wortfeld mit assoziierten Lexemen angezeigt und es folgen Beispielsätze aus den DWDS-Korpora. Oben rechts im Bild ist außerdem eine Frequenzkurve sichtbar, die die relative Worthäufigkeit (pro Millionen Token im zugrunde liegenden Korpus) auf einer Zeitachse von ca. 1600 bis 2000 anzeigt. Zuletzt sind darunter Korpusressourcen des DWDS aufgelistet, in denen das gesuchte Wort auftritt.

V1993

Die beschriebenen Elemente dieser Zusammenstellung (Wörterbucheinträge, Korpusdaten und statistische Ausgaben) werden vom DWDS auch einzeln angeboten: Man kann die Oberbereiche »Wörterbücher« (http://www.dwds.de/wb), »Korpora im DWDS« (http://www.dwds.de/r) und »Statistische Auswertungen« (http://www.dwds.de/stats) gesondert anwählen. Im Wörterbuchbereich werden verschiedene Wörterbuchressourcen vorgestellt, aus denen sich das Wörterbuch des DWDS speist und von denen das »Wörterbuch der deutschen Gegenwartssprache« und »Das Deutsche Wörterbuch von Jacob Grimm und Wilhelm Grimm (Erstbearbeitung)« getrennt abfragbar sind.

Ist ein bestimmter Suchbegriff nicht im Wörterbuch enthalten, wird dies angezeigt. Dennoch stellt das System dann bestimmte wortgrammatische Informationen zusammen, die entsprechend automatisch erhoben werden.

In den kommenden Abschnitten werden die korpuslinguistisch relevanten Funktionen des DWDS beschrieben.

Internet-Links für die Korpussuche: Um die Suche in den Korpora DWDS und DTA nachvollziehen zu können, besuchen Sie bitte die Webseiten http://www.dwds.de/r/ bzw. http://www.deutschestextarchiv.de/.

Auswahl von Korpusdaten: Im Gegensatz zu den meisten Suchinterfaces muss im DWDS und DTA kein Korpus ausgewählt werden, da im DWDS eine Standardressource durchsucht wird (das sog. DWDS-Kernkorpus mit gut sieben Milliarden Token) und im DTA das gesamte Korpus »Deutsches Textarchiv«, das im DWDS-Interface mit 210858587 Token und auf der Webseite des Deutschen Textarchivs mit »154914194 fortlaufende Wortformen« angegeben wird.

Im DWDS-Interface kann man (ohne Anmeldung als registrierte Nutzerin oder registrierter Nutzer) per Auswahlmenü zwischen 12 verschiedenen Korpora auswählen, deren jeweilige Gesamtgröße in einer separaten Zusammenstellung aufgeführt wird. Die nachfolgenden Suchmöglichkeiten beziehen sich auf alle auswählbaren Ressourcen.

Suche nach Lemmata: Die Wortformsuche ist komplizierter als die Lemmasuche, da die Eingabe eines Suchbegriffs wie *Hund* automatisch nach der Grundform sucht und somit auch Suchtreffer wie *Hunde* und *Hundes* hervorbringt. Auch die Eingabe flektierter Formen wie *Hundes* liefert alle Formen des Paradigmas, einschließlich *Hund*. Der Grund liegt in der Tatsache, dass das DWDS in erster Linie als Lexikon konzipiert ist. Sämtliche Formen eines Paradigmas sollen zu demselben Eintrag führen. Somit muss die Nutzerin oder der Nutzer nicht überlegen, mit welcher Form ein bestimmter Eintrag repräsentiert ist; z. B. könnte das Indefinitpronomen *anderer* auch mit den Formen *andere* oder *ander* lemmatisiert sein. In dem konkreten Fall führen alle zugehörigen Formen (z. B. auch *anderes*) zu dem Eintrag *ander*. Geben Sie z. B. den Begriff

```
laufe
```

ein, um alle Formen des Verbs *laufen* zu erhalten.

Suche nach Wortformen: Will man eine bestimmte Form finden, ohne

weitere zum Lemma gehörige Formen ausgegeben zu bekommen, so muss man der gesuchten Form ein @-Zeichen voranstellen. Die Suche

```
@Hundes
```

findet also ausschließlich die Form *Hundes*, einschließlich Groß- und Kleinschreibung.

Suchen nach Wortarten erfolgen mittels der Kategorien des STTS. Diese werden ohne Anführungszeichen hinter den Ausdruck *$p=* gesetzt, der die Suchvariable »Wortart« spezifiziert. Die Suche

```
$p=APZR
```

findet somit alle als Postpositionen annotierten Einheiten im Korpus. Schauen Sie unter dem Menüpunkt »Suchen nach mehreren Eigenschaften desselben Token«, um zu erfahren, wie die Wortartensuche mit der Wortformsuche oder Lemmasuche kombiniert werden kann.

Suchen mit regulären Ausdrücken werden wie folgt unterstützt:

- Das Sternchen an den Rändern des Suchausdrucks drückt eine beliebige Zeichenkette aus. Hierdurch werden nur dem Suchausdruck entsprechende Wortformen gefunden:
  ```
  *wand
  ```
  findet alle Wortformen, die auf *-wand* enden.
- Mit in Schrägstrichen (»/«) um den Suchausdruck herum bezieht sich die Suche auf Lemmata. Hierbei sind reguläre Ausdrücke so zu verwenden, wie in Kapitel 3.1.2.8f. beschrieben:
  ```
  /.*igkeit/
  ```
  findet Nomina, deren Lemma auf *-igkeit* endet. Es werden also auch Formen wie *Schwierigkeiten* gefunden.
- Die Markierung mit Schrägstrichen gilt auch für die Wortartensuche. So findet der Ausdruck
  ```
  $p=/P.*/
  ```
  alle Wörter, die laut dem STTS Pronomina sind.
- Alternative Elemente, egal, ob Lemmata, Wortformen oder Wortarten, können durch Kommata getrennt in geschweiften Klammern aufgeführt werden:
  ```
  @{Leute,Menschen}
  ```
  findet genau die Formen *Leute* oder *Menschen*;
  ```
  {wollen,mögen,möchten}
  ```
  findet alle Formen der Lemmata *wollen*, *mögen* und *möchten*;
  ```
  $p={ADV,NN}
  ```
  findet Adverbien oder Nomina.

Suche nach Abfolgen: Direkte Abfolgen werden durch Nebeneinanderstellen verschiedener Suchelemente (wie in der CQP-Syntax) ausgedrückt. Die gesamte Abfolgesuche wird in Anführungszeichen gesetzt. So findet der Suchausdruck

```
"$p=/(P.*AT|ART)/ $p=ADJA $p=NN"
```

alle unmittelbaren Abfolgen von Artikelwörtern, Adjektiven und Nomina. Der Suchausdruck

```
"@{Leute,Menschen} {wollen,mögen,möchten} $p={ADV,NN}"
```

findet unmittelbare Abfolgen der Oberflächenform *Leute* oder *Menschen*, gefolgt von einer Form von *wollen*, *mögen* oder *möchten*, gefolgt von einem Adverb oder einem Nomen (gefunden wird z. B. *Menschen wollten Frieden*).

Um einen Maximalabstand bzw. Abstandsbereich zu definieren, wird zwischen dem präzedenten Ausdruck und dem nachfolgenden Element die maximale Anzahl von dazwischenstehenden Elementen nach dem Rautenzeichen definiert:

```
"@sein #3 Hund"
```

findet Folgen der Wortform *sein* und dem Lemma *Hund*, wobei bis zu drei Elemente dazwischen stehen dürfen. Für den exakten Abstand wird zwischen Raute und Zahl ein Gleichheitszeichen eingefügt; analog gilt dies für den Ausdruck »größer als« (" > ") und »kleiner als« (" < ").

Suche innerhalb desselben Satzes: Da die DWDS-Korpusdaten eine Satzsegmentierung besitzen, kann man nach Elementen innerhalb desselben Satzes suchen. Hierzu verwendet man den Zusatz »&&«, gefolgt von dem Element, das innerhalb desselben Satzes existent sein soll. So findet die Suche

```
"@Menschen wollen" && {ja,wohl}
```

die Abfolge der Wortform *Menschen*, gefolgt von einer Form von *wollen*, und innerhalb desselben Satzes tritt entweder *ja* oder *wohl* auf. Ebenso können Elemente innerhalb eines Satzes ausgeschlossen werden. (Der Negationsoperator ist das Ausrufezeichen.) Z. B. findet die Suche

```
"@Alle $p=NN wollen" && !$p=ADJA
```

Abfolgen von *Alle*, einem Nomen und dem Lemma *wollen*, ohne dass in demselben Satz ein pränominales Adjektiv vorkommt. Wörter können durch einfaches Voranstellen des Ausrufezeichens negiert werden:

```
"@Lügen @haben" && !kurz
```

findet die Abfolge *Lügen* und *haben*, ohne dass in demselben Satz das Lemma *kurz* auftritt.

Bezogen auf den Satzanfang kann für ein gegebenes Element eine bestimmte Position angegeben werden, wobei nach dem Element der Zusatz "WITH $.=" und eine Zahl, die den Abstand zum Satzanfang spezifiziert, hinzugefügt wird. Der Ausdruck

```
$p=VMFIN WITH $.=0
```

findet auf diese Weise finite Vollverben am Satzanfang; der Ausdruck

```
$p=/V.FIN/ WITH $.=1
```

findet finite Verben genau nach einem Element ab dem Satzanfang usw.

Suche nach mehreren Eigenschaften desselben Token: Eine gleichzeitige Suche nach mehreren Eigenschaften desselben Elements im Korpus – z. B. sein Lemma und seine Wortart – wird mittels des Operators »with« ausgedrückt. Sucht man z. B. alle Vorkommen der Wortform *aus*, die Verbpartikeln sind, so lautet die DWDS-Suchanfrage

```
@aus with $p=PTKVZ
```

Um ~~(z. B. im Deutschen Textarchiv, das nicht normierte Schreibungen enthält, die einheitlich lemmatisiert sind) alle Wortformen zu finden, die~~ demselben Lemma und derselben Wortart ~~zugeordnet sind~~, lassen Sie das @-Zeichen weg:

```
schwarz with $p=ADJA
```

findet alle Wortformen (wie *schwarzem* oder *schwarzes*) mit dem Lemma *schwarz*, die als pränominale Adjektive verwendet werden.

Suche nach allen Instanzen einer Variable: Im DWDS- und DTA-System ist es nicht möglich, sämtliche Vorkommen einer bestimmten Variable (z. B. alle Wortarten oder Lemmata) zu suchen, um z. B. unkompliziert Frequenzlisten der einzelnen Typen zu erstellen. Für Möglichkeiten, die andere Korpuswerkzeuge hinsichtlich dessen bieten s. Kapitel 4.3.

Quantifizierbarkeit der Ergebnisse: Für absolute Frequenzen (die Gesamttrefferzahl zu den durchgeführten Suchen werden jeweils angegeben) können relative bzw. normalisierte Frequenzen (s. Kap. 4.5.1) ermittelt werden, weil die Größe der durchsuchten Korpusdaten in der Korpusbeschreibung ersichtlich ist: Die Tokenzahlen pro Korpus sind dokumentiert. Weitere Gesamtgrößen (z. B. die Anzahl von Wortarten in einem bestimmten Korpus) können durch Korpussuchen ermittelt werden. Verschiedene sogenannte »Count-Abfragen« ermitteln unmittelbar Frequenzen bzw. Frequenzlisten, sortiert nach verschiedenen Parametern wie z. B. Wortformen zu einem Lemma oder Vorkommen pro Jahrzehnt (~~http://~~www.dwds.de/d/suche#korpussuche, unten). Beispiele hierfür sind:

- Geben Sie den Suchausdruck
  `count( Hund ) #by[$w] #DESC_COUNT`
  in das Suchfeld ein und Sie erhalten alle Formen des Lemmas *Hund*, nach Häufigkeit im Korpus absteigend sortiert.
- Geben Sie den Suchausdruck
  `count( Bibel ) #by[date/10]`
  in das Suchfeld ein und Sie erhalten eine Tabelle mit den Häufigkeiten des Lemmas *Bibel* in verschiedenen Jahrzehnten innerhalb des Zeitspektrums, das das Korpus abdeckt. Hinweis: Geben Sie beim Vergleichen dieser Werte Acht, da die aufgelisteten Häufigkeiten nicht immer direkt miteinander vergleichbar sind (zur Normalisierung von Fre-

quenzen s. Kap. 4.5.1). ~~[Tipp: Verwenden Sie diesen und den folgenden Befehl für verschiedene Korpora des DWDS.]~~

- Geben Sie den Suchausdruck
  `count( aus ) #by[$p] #DESC_COUNT`
  in das Suchfeld ein und Sie erhalten eine Tabelle mit den Häufigkeiten des Lemmas *aus*, frequenzmäßig absteigend sortiert nach den verschiedenen Wortarten, die *aus* haben kann.
- Geben Sie den Suchausdruck
  `count (aus with $p=PTKVZ) #by[$w] #DESC_COUNT`
  in das Suchfeld ein und wählen Sie als zu durchsuchendes Korpus das Deutsche Textarchiv aus, um alle dort enthaltenen historischen Schreibungen von *aus* als Verbpartikel, absteigend sortiert nach ihrer Häufigkeit im Korpus, aufzulisten.

Die Exportmöglichkeiten des DWDS-Interfaces sind relativ vielseitig. Angeboten werden die Exportformate CSV, JSON, TCF und TSV (zur Relevanz dieser Formate s. Kap. 4.1). Der Export ist jedoch auf maximal 5000 Treffer beschränkt. Metadaten können mit den herausgelesenen Treffern gemeinsam exportiert werden.

Arbeitsaufgaben

1. Formulieren Sie eine Suche im DWDS-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.
2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.
3. Testen Sie diese Suchen an dem »DWDS-Kernkorpus (1900–1999)« in der Funktion der DWDS-Korpusbelege (~~http://~~www.dwds.de/r).

3.2.2 | COSMAS II (und KorAP) des IDS

Das COSMAS-System des Instituts für Deutsche Sprache (IDS; ~~http://~~www.ids-mannheim.de/cosmas2/) ist das wahrscheinlich bekannteste Online-Suchsystem für Korpora. Es befindet sich in der zweiten Generation. Aktuell wird ein Nachfolgeprodukt KorAP erstellt (~~http://~~www1.ids-mannheim.de/kl/projekte/korap), das erweiterte Such- und Exportfunktionen bieten wird. Was die grundlegenden Funktionen angeht, die auf die verschiedenen existierenden IDS-Korpora zugeschnitten sind, so wird die Bedienung der Systeme ~~beid~~ vergleichbar sein. Nachfolgend werden die wesentlichen Suchfunktionen in COSMAS II vorgestellt.

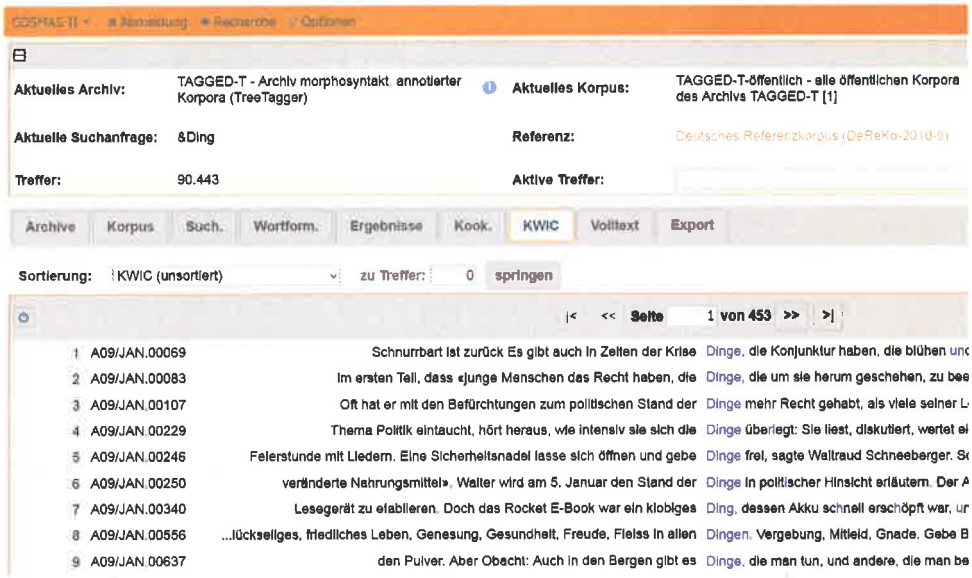


Abb. 3.11: Das COSMAS-Suchinterface (Registrierung/Login erforderlich: http://cosmas2.ids-mannheim.de/cosmas2-web/)

Die Auswahl von Korpusdaten aus einer Gesamtmenge zur Verfügung stehender Teilkorpora ist eine der Kernideen des Interfaces: Im Gegensatz zu den meisten Suchinterfaces kann man in COSMAS II und KorAP beliebige Korpora auswählen und Suchanfragen über diese Auswahl stellen können. ~~COSMAS II bietet aber auch Standardmöglichkeiten zum Durchsuchen von Korpora ohne individuelle Korpuszusammenstellung.~~ Der allgemeine Weg zur Auswahl der zu durchsuchenden Korpusressourcen bis hin zur Sichtung von Suchergebnissen sei an dem folgenden COSMAS-Screenshot erläutert.

Abb. 3.11 zeigt das COSMAS-Suchinterface zum Zeitpunkt der Trefferssichtung, also nach einer bereits gestellten Suchanfrage. Ab dem Login (dies erfolgt auf der oben genannten Webseite unter dem Menüpunkt

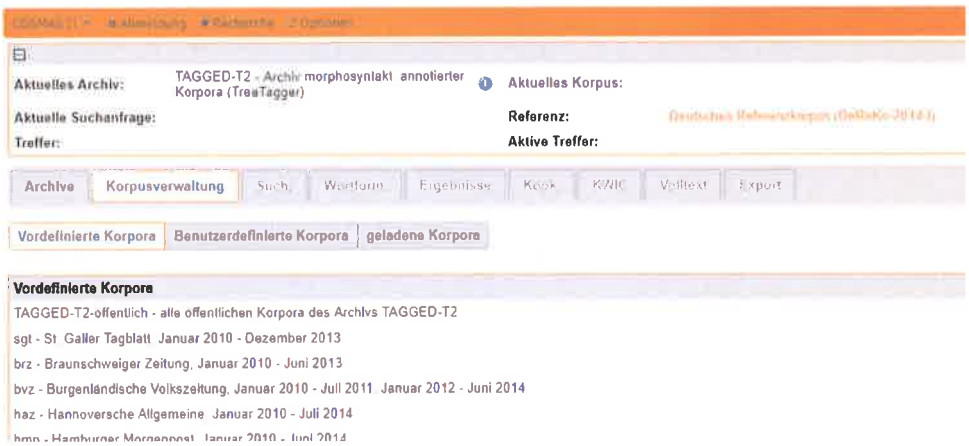


Abb. 3.12: Menüpunkt »Korpusverwaltung« im COSMAS-Suchinterface

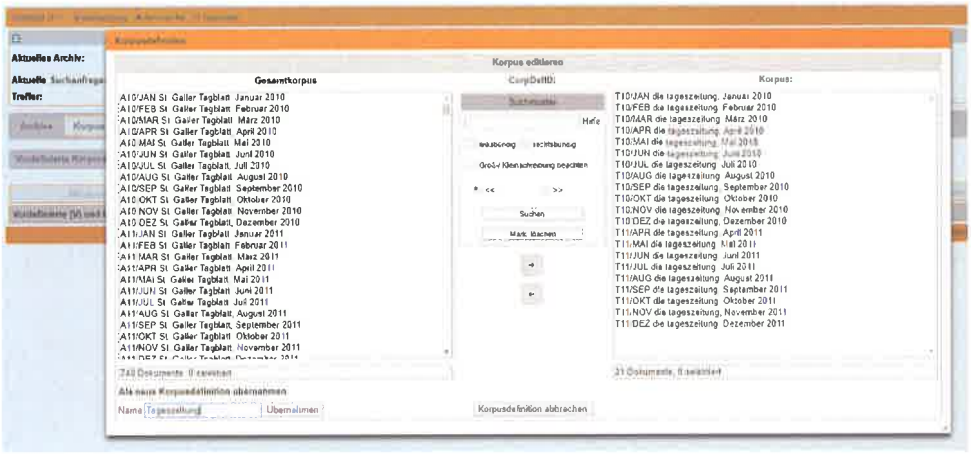


Abb. 3.13: Die Auswahl individuell zusammengestellter Korpora aus dem Gesamtrepertoire des COSMAS-Systems

»Anmeldung« > »Login«) arbeitet sich die Nutzerin oder der Nutzer im Grunde genommen durch die horizontale, graue Menüleiste ab dem Menüpunkt »Archive«. Hier werden größere Zusammenstellungen vergleichbarer Korpora zur Auswahl bereitgestellt. Nachdem man eine Auswahl getroffen hat, kann man unter dem nachfolgenden Menüpunkt »Korpus« (bzw. »Korpusverwaltung«) ein einzelnes in dem Archiv enthaltenes Korpus ~~oder auch~~ verschiedene Korpora oder das gesamte Archiv auswählen.

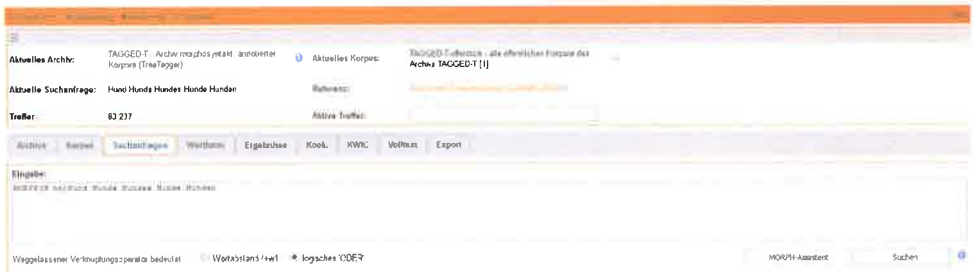
Alternativ kann man aus dem Gesamtrepertoire der verfügbaren Korpora eine individuelle Zusammenstellung vornehmen. Hierzu ist es für die bessere Vergleichbarkeit der einzelnen Korpora notwendig, sich auf der Dokumentationsseite ~~http://~~www.ids-mannheim.de/cosmas2/projekt/referenz/ mit der Beschaffenheit der einzelnen Ressourcen vertraut zu machen. Individuelle Korpuszusammenstellungen können unter dem Menüpunkt »geladene Korpora« > «Neu« vorgenommen werden.

Die Erläuterungen zu den folgenden Suchfunktionen sind anhand der Korpuszusammenstellung »TAGGED-T« (»TAGGED-T-öffentlich – alle öffentlichen Korpora des Archivs TAGGED-T«) aufbereitet worden.

Abb. 3.14: Suchoberfläche mit Sucheingabefenster und Suchoptionen in COSMAS II

Vom Abschicken der Suche zur Ansicht der Treffer: Wenn Sie eine Suchanfrage formuliert ~~haben~~ und geeignete Optionen für die Suche ausgewählt haben, schicken Sie die Suche mithilfe der »Suchen«-Funktion rechts unter dem Sucheingabefenster ab.

Anschließend erhält man eine Rückmeldung zu den Formen der ausgewählten Korpusdaten, die durch die eingegebene Suchanfrage gefun-



den wurden. Um die Ergebnisse einsehen zu können, muss man zunächst die Auswahlfunktion »Ergebnisse« unten links anklicken. In dem nun erscheinenden Ergebnisfenster sind die Suchergebnisse nach verschiedenen Quellen (Subkorpora) getrennt und können einzeln per Mausklick geöffnet werden. Um sämtliche Ergebnisse zusammen einsehen zu können, klicken Sie in der mittleren grauen Auswahlleiste auf »KWIC« (key word in context), um die Treffer mit verhältnismäßig wenig Zusatzinformationen zu betrachten, oder wählen Sie »Volltext«, um mehr Metadaten zu den Treffern sehen zu können und den Kontext um die Treffer größer stellen zu können. Den angezeigten Trefferkontext kann man unter »Optionen« variieren.

Suche nach Wortformen: Die Exaktheit einer eingegebenen Zeichenkette, z. B. »Hundes«, wird nicht über den Suchausdruck selber festgelegt (wie in allen zuvor beschriebenen Korpussuchsystemen), sondern wird über Suchoptionen in einem gesonderten Menü festgelegt: Klicken Sie in der oberen orangefarbenen Menüleiste auf die Funktion »Optionen«. Wenn Sie in dem erscheinenden Optionsmenü die ersten drei Auswahloptionen aktivieren, wird die Suche so exakt wie möglich durchgeführt, d. h. die Klein- und Großschreibung der Zeichen im Suchausdruck wird berücksichtigt und nicht-alphabetische Zeichen innerhalb der Treffer werden weitestgehend ausgeschlossen. Es lässt sich in COSMAS II aber grundsätzlich nicht vermeiden, dass Satzzeichen, die mit dem gesuchten Ausdruck verschmolzen sind (weitestgehend Tokenisierungsfehler) und Bindestriche ausgeschlossen werden.

Geben Sie also den Suchbefehl

```
Hundes
```

in das Eingabefenster unter dem Menüpunkt »Suchanfragen«, stellen Sie in den »Optionen« die Suchmodalitäten so ein, dass alle drei Beschränkungen aktiviert sind, und schicken Sie die Suche ab, um Ergebnisse möglichst ausschließlich für die Wortform *Hundes* zu erhalten. Sie müssen anschließend die Funktion »Ergebnisse« betätigen und im mittleren grauen Auswahlmenü die Option »KWIC« auswählen, um alle Treffer im Kontext betrachten zu können.

Suche nach Lemmata: Um sämtliche Formen, die demselben Lemma zugewiesen sind, finden zu können, markiert man die gesuchte Zeichenkette mit einem vorangestellten &-Zeichen. Geben Sie also

```
&Hund
```

in die Suchmaske ein, um alle Formen zu finden, die mit *Hund* lemmatisiert wurden.

Suche nach Wortarten: COSMAS II verwendet ein individuelles morphosyntaktisches Tagging, das aus dem STTS abgewandelt ist und teilweise zusätzliche Informationen enthält. Verwenden Sie den »Morph-Assistent«, um das für Ihre Suche geltende Wortartentag zu ermitteln. Nehmen wir z. B. an, wir wollen alle Verbpartikeln im Korpus ermitteln. Dann wird mittels des Morph-Assistenten über die Hauptkategorie »Partikel«

und die Unterkategorie »abgetrennter Verbzusatz« nicht nur der Wert »PTK vz« ermittelt, sondern gleich innerhalb eines verwertbaren Suchausdrucks in das Eingabefenster überführt:

```
MORPH(PTK vz)
```

findet alle Verbpartikeln im Korpus. An diesem Suchausdruck sieht man, dass die COSMAS-Anfragesprache Suchvariablen wie »MORPH« für »morphosyntaktische Suchebene« dem Wert voranstellt, der wiederum in runde Klammern gesetzt wird.

Wenn Sie diese Suchbedingung mit einer Wortform oder einem Lemma verknüpfen wollen, lesen Sie weiter unten unter dem Punkt »Suche nach mehreren Eigenschaften desselben Token« weiter.

Suche mit regulären Ausdrücken: In COSMAS II werden reguläre Ausdrücke unterstützt, allerdings nicht mit der Standardbedeutung der verschiedenen Operatoren (s. Kap. 3.1.2.17). Mustersuchen können mit folgenden Operatorenbedeutungen durchgeführt werden:

- Der Sternchenoperator »*« steht für beliebig viele Zeichen (er entspricht also dem Ausdruck ».*« in den meisten Suchsystemen mit regulären Ausdrücken.
- Der Fragezeichenoperator »?« bedeutet »ein beliebiges Zeichen« und entspricht somit dem Ausdruck ».« in den meisten Suchsystemen mit regulären Ausdrücken.
- Der Plusoperator »+« steht für ein beliebiges oder kein Zeichen und entspricht somit dem Ausdruck ».?« in den meisten Suchsystemen mit regulären Ausdrücken.

Somit findet die Suche

```
*barkeit
```

alle Wörter im Korpus, die auf *-barkeit* enden. Die Suche

```
+++++barkeit
```

findet auch Wörter, die auf -barkeit enden, allerdings nur solche mit maximal fünf Zeichen vor der Endung. Die Suche

```
*hunger*
```

findet alle Wörter, die das lexikalische Morphem *hunger* bzw. *Hunger* enthalten, einschließlich *Hunger* selbst.

Alternativen werden mit dem Operator »ODER« verknüpft. Die Suchanfrage

```
&Hund ODER &Katze
```

findet demnach die Lemmata *Hund* und *Katze*. Wenn man allerdings die Sucheinstellung »Weggelassener Verknüpfungsoperator bedeutet ... logisches ›ODER‹« auswählt, kann man den Operator auch weglassen.

Suchen nach Abfolgen zwischen Elementen (Wortformen, Lemmata oder Wortarten) werden mithilfe eines Operators zwischen den Elementen gekennzeichnet. Für unmittelbare Abfolgen verwendet man den Ausdruck »/w1«, wobei die Zahl die Größe des Abstands ausdrückt und die Reihenfolge der links und rechts vom Abstandsausdruck stehenden Elemente beliebig ist. Für eine festgelegte Reihenfolge des linken vor dem rechten Element verwendet man ein zusätzliches »+«-Symbol: »/+ w1«. Auf diese Weise findet man mit dem Suchausdruck

```
&ein /+w1 MORPH(ADJ at) /+w1 &Vorschlag
```

Abfolgen des unbestimmten Artikels *ein* (als Lemma), gefolgt von einem attributiven (pränominalen) Adjektiv, gefolgt von einer Form des Nomens *Vorschlag*. Bei größerem Abstand gibt man mit einem Zahlenwert den Höchstabstand ein. So findet man mit dem Ausdruck

```
MORPH(AP pr) /+w3 MORPH(N)
```

Präpositionen und Nomina mit einem Maximalabstand von drei Stellen.

Suche nach mehreren Eigenschaften desselben Token: Um zu erreichen, dass sich mehrere Suchbedingungen auf dasselbe Element im Korpus beziehen, verwendet man den Wortabstandsoperator mit dem Abstand 0: »/w0«.

```
Entgegen /w0 MORPH(PTK vz)
```

findet auf diese Weise alle Vorkommen der Wortform *Entgegen* (also satzinitial), die als Verbpartikeln analysiert sind. Aktivieren Sie hierfür vor dem Abschicken der Suche in den »Optionen« die Unterscheidung von Groß- und Kleinschreibung am Wortanfang. Ohne diese Einstellung zu berücksichtigen, kann man mit den Anfragen

```
trotz /w0 MORPH(AP pr)
```

und

```
trotz /w0 MORPH(N nn)
```

den präpositionalen Gebrauch von *trotz* (bzw. *Trotz* am Satzanfang) und den nominalen Gebrauch von *Trotz* auseinanderhalten.

Suche nach allen Instanzen einer Variable: Wie im DWDS- und DTA-System ist es in COSMAS II nicht möglich, sämtliche Vorkommen einer bestimmten Variable (z. B. alle Wortarten oder Lemmata) zu suchen, um z. B. unkompliziert Frequenzlisten der einzelnen Typen zu erstellen. Für andere Korpuswerkzeuge, die diese Funktion bieten, s. Kapitel 4.3.

Quantifizierbarkeit der Ergebnisse: Da die Größe der durchsuchten Gesamtdatenmenge in COSMAS II stets bekannt ist (Angaben in Wörtern für die geladenen Korpora unter dem Menüpunkt »Korpusverwaltung« in der zentralen grauen Menüleiste) und auch die Gesamtzahl der Treffer zu

einer Suche angezeigt wird, ist es möglich, relative bzw. normalisierte Werte (s. Kap. 4.5.1) zu ermitteln. Auch bietet das System durch seine Exportfunktion, in der diverse Parameter eingestellt werden können, die Möglichkeit, Frequenzlisten mit Wortform- oder Lemmahäufigkeiten zu exportieren.

Exportmöglichkeiten: Die durch eine Suchanfrage erzielten Treffer können mithilfe der Funktion »Export« (ganz rechts im mittleren grauen Auswahlmenü) als Textdatei herausgeschrieben werden, wobei die maximale Anzahl der einsehbaren Treffer auf 10.000.000 reduziert ist. Im Fall einer höheren Trefferzahl wird die Gesamttrefferzahl genannt und aus dieser Zahl die maximale Anzahl der anzeigbaren Treffer zufällig gezogen. Hierbei kann der Trefferkontext reguliert werden, verschiedene Exportformate (KWIC und Volltext mit oder ohne Annotationen) können ausgewählt und miteinander verknüpft werden sowie Frequenzlisten zu festgelegten Parametern (z. B. nach Zeitintervallen oder Wort-Typen) erstellt werden.

Arbeitsaufgaben

1. Formulieren Sie eine Suche im COSMAS-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.

2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.

3. Testen Sie diese Suchen an dem TAGGED-T-Archiv des COSMAS-II-Interfaces.

3.2.3 | DGD des IDS

»DGD« steht für »Datenbank für gesprochenes Deutsch«. Hierbei handelt es sich um ein Suchinterface für die durch das Institut für Deutsche Sprache bereitgestellten ~~Korpora der gesprochenen Sprache~~ Gesprächskorpora. Die Webseite der DGD ist https://dgd.ids-mannheim.de/. Um Korpussuchen durchführen zu können, muss man sich wie in COSMAS II registrieren.

Im Interface kann man wie in COSMAS II Korpusressourcen auswählen und aggregieren (linkes Auswahlmenü). Das Kernstück der Korpussuche ist die tokenbasierte Recherche in den Transkripten und Annotationen individuell ausgewählter Korpusdaten. Hierzu wählt man im oberen horizontalen Menü den Punkt »Recherche« und dort die Funktion »Tokens« aus. Standardmäßig ist das FOLK-Korpus (http://agd.ids-mannheim.de/folk.shtml) ausgewählt.

Die folgenden Suchbeispiele beziehen sich auf die in Abb. 3.15 gezeigte Auswertungseinstellung im DGD-Suchinterface (»Recherche« >

Abb. 3.15: Die DGD-Nutzeroberfläche mit aktivierter Korpusrecherche auf Token-Ebene

»Tokens«). Im Gegensatz zu den bisher behandelten Korpussuchsystemen erfolgt die Angabe der Annotationsebene nicht mittels Suchausdruck, sondern durch vorgegebene Eingabefelder. Dies erleichtert die Suche für unerfahrene Nutzerinnen und Nutzer, auf der anderen Seite sind die Suchmöglichkeiten dadurch jedoch stark eingeschränkt.

Suche nach transkribierten Wortformen: Will man gewisse dialektale oder umgangssprachliche Wortformen finden, so kann man das Eingabefeld »Transkribiert« nutzen. Die hier erfragbaren Formen sind nach dem Minimaltranskript des GAT2-Transkriptionssystems transkribiert worden (s. vor allem Kap. 2.4.4). So kann man z. B. mit der Anfrage

Eingabefeld »Transkribiert«: `ham`

Vorkommen der ersten und dritten Person Plural von *haben* finden, die *ham* gelautet wurden. Beachten Sie, dass auf der Transkriptionsebene keine Groß- und Kleinschreibung unterschieden wird (Nomina werden kleingeschrieben). Deshalb findet z. B. die Suche

Eingabefeld »Transkribiert«: `lauf`

sowohl Vorkommen gebeugter Formen des Verbs *laufen* am Satzanfang und innerhalb von Sätzen sowie das Nomen *Lauf*.

Suche nach normalisierten Wortformen: Die mit der literarischen Umschrift transkribierten Wortformen werden auf einer Normalisierungsebene im Korpus der deutschen Standardorthographie angepasst (inklusive der Substantivgroßschreibung, aber ohne die satzinitiale Großschreibung und ohne Interpunktion). Auf diese Annotationsebene greift man über das Eingabefeld »Normalisiert« zu. So findet die Suche

Eingabefeld »Normalisiert«: `haben`

sämtliche Vorkommen des Verbs *haben* in der normalisierten Form *haben*, unabhängig von deren lautlicher Realisierung (nicht gefunden wer-

den hier finite Formen von *haben*, abgesehen von der ersten und dritten Person Plural). Die Suche

Eingabefeld »Normalisiert«: `Laut`

findet ausschließlich Vorkommen des Nomens *Laut* mit der normalisierten Form *Laut* (nicht gefunden werden hier Genitivformen und Pluralformen).

Suche nach Lemmata: ~~Auf der Annotationsebene der Lemmata kann man nach Grundformen suchen.~~ Auch ~~hierbei~~ wird die standarddeutsche Orthographie inklusive der Substantivgroßschreibung berücksichtigt. Das Eingabefeld für Lemmasuchen lautet »Lemma«. Mit der Suche

Eingabefeld »Lemma«: `haben`

findet man sämtliche Formen des Verbs *haben* unabhängig von ihrer lautlichen Realisierung.

Die Suche nach Wortarten in den DGD-Korpora basiert im Kern auf dem STTS-Tagset (s. Kap. 2.2.6.1), umfasst aber z. B. im FOLK-Korpus einige weitere Tags, die Phänomene der Mündlichkeit beschreiben. Rechts neben dem Eingabefeld für die Wortarten, »POS«, kann man per Mausklick je nach aktiviertem Korpus eine Auflistung der verfügbaren Tags öffnen. (Hinweis: Die Interpunktionszeichen des STTS sind in der Auflistung zwar enthalten, diese wurden jedoch nicht vergeben, weil auf der Normalisierungsebene der Korpora keine Satzzeichen hinzugefügt wurden.) So kann man mit Suchen wie

Eingabefeld »POS«: `PTKVZ`

alle abgetrennten Verbpartikeln im Korpus finden, wie es für STTS-getaggte Korpora üblich ist. Zusätzlich findet man mit Suchen wie

Eingabefeld »POS«: `SEDM`

alle als Diskurspartikeln interpretierten Wörter im Korpus (diese Klasse wird im Standard-STTS der Wortklasse ADV untergeordnet).

Die Suche mit regulären Ausdrücken wird unterstützt, wie in Kapitel 3.1.2 bzw. zusammengefasst in Kapitel 3.1.2.17 beschrieben. Um reguläre Ausdrücke als solche interpretiert zu bekommen, muss man die Option »Reguläre Ausdrücke« aktivieren. Nun wird z. B. die Suche

Eingabefeld »Normalisiert«: `(könntest|müsst.*)`

so verarbeitet, dass die normalisierten Wortformen *könntest* und *müsst-* sowie mit *müsst-* beginnende Wörter gefunden werden.

Die Suche nach Abfolgen erfolgt, indem zunächst die Suche nach einem Element auf der Ebene »Token« (im zweiten horizontalen Auswahlmenü) durchgeführt wird und die daraus resultierende Treffermenge dann anschließend durch zusätzliche Eingaben unter dem Menüpunkt »Kontext« weiter eingeschränkt wird. Will man z. B. unmittelbare Abfol-

gen einer finiten Form von *haben* und einem Personalpronomen finden, so gibt man unter »Token« zunächst ein:

Eingabefeld »Normalisiert«: `haben`
Eingabefeld »POS«: `V.FIN`
(Hinweis: Die Option »Reguläre Ausdrücke« muss aktiviert sein.)

Anschließend klickt man weiter auf »Kontext« und gibt dort die Suchverfeinerung ein:

Eingabefeld »POS«: `PPER`

Nun wählt man rechts bei den zwei Auswahlmenüs zu »Kontext« die Werte »1 Token« und »rechts« aus. Schickt man diese Verfeinerung mit einem Klick auf die Option »Kontext filtern« ab, so werden die Treffer, auf die die zweite Suchrestriktion nicht mehr zutrifft, ausgeblendet.

Suche nach mehreren Eigenschaften desselben Token: Durch die gleichzeitige Nutzung der verfügbaren Eingabefelder kann man verschiedene Merkmale gleichzeitig berücksichtigen. Kombiniert man die Eingabefelder »Transkribiert« und »Normalisiert«, so kann man z. B. mit der folgenden Suche Fälle finden, in denen Wörter, die orthographisch auf *-gt* enden, spirantisiert artikuliert werden:

Eingabefeld »Transkribiert«: `.*cht`
Eingabefeld »Normalisiert«: `.*gt`
(Hinweis: Die Option »Reguläre Ausdrücke« muss aktiviert sein.)

Suche nach allen Instanzen einer Variable: Auch in der DGD ist es nicht möglich, sämtliche Vorkommen einer bestimmten Variable (z. B. alle Wortarten oder Lemmata) zu suchen, um z. B. unkompliziert Frequenzlisten der einzelnen Typen zu erstellen. Für Möglichkeiten, die andere Korpuswerkzeuge hinsichtlich dessen bieten, s. Kap. 4.3.

Quantifizierbarkeit der Ergebnisse: Mit der Liste der Treffer zu einer gegebenen Suchanfrage erhält man zunächst eine Gesamttrefferzahl. In dem horizontalen Menü direkt oberhalb der Treffer verbirgt sich hinter einem Symbol, das ein Säulendiagramm zeigt, eine Quantifizierungsoption. Hier erhält man die Gesamtmenge der durchsuchten Daten (in Token) sowie verschiedene Angaben zur Anzahl verschiedener Typen (z. B. die Anzahl der gefundenen Lemmata zu einer bestimmten Mustersuche), der Anzahl durchsuchter Transkripte usw.

Exportmöglichkeiten: Die ermittelten Treffer zu einer Korpussuche können als Textdatei oder auch als XML-Datei exportiert werden. Statistische Werte kann man exportieren, indem man die in einem separaten Browserfenster erzeugten quantifizierten Daten ~~[s. o.]~~ in einen Editor kopiert.

Arbeitsaufgaben

1. Formulieren Sie eine Suche im DGD-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.
2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.
3. Testen Sie diese Suchen an dem FOLK-Korpus im Suchinterface.

3.3 | Evaluation von Korpussuchen

Der Erfolg von Korpussuchen wird durch zwei Fragen bemessen:

- Entsprechen die durch die Suche gefundenen Treffer der Erwartung: Gehören sie zu der Kategorie, die gefunden werden sollte? (Diese Frage beantwortet die Berechnung des Werts ›Precision‹, s. Kap. 3.3.1.)
- Wurden alle Elemente, die hätten gefunden werden sollen, auch gefunden? Wenn nicht: Wie groß ist die Anzahl der verpassten Elemente? (Diese Frage beantwortet die Berechnung des Werts ›Recall‹, s. Kap. 3.3.2.)

Je nach der der Korpusrecherche zugrunde liegenden Fragestellung bzw. dem Ziel der Korpussuche sind die Antworten auf diese Fragen ganz unterschiedlich zu interpretieren. Ist es z. B. das Anliegen einer Korpussuche, eine Liste von Beispielen zu einer bestimmten linguistischen Struktur (z. B. Sätze ohne Objekt, aber mit freiem Dativ) zu ermitteln, ohne dass die Gesamtheit aller im Korpus enthaltenen Strukturen erfasst werden muss, so spielt die erste Frage eine entscheidende Rolle, die zweite aber nicht. Soll aber eine Suchanfrage dazu dienen, sämtliche im Korpus enthaltene Treffer zu einer Struktur (z. B. Relativsätze) manuell in Unterklassen (z. B. restriktive vs. appositive vs. freie Relativsätze) einzuteilen, so ist entscheidend, dass alle Fälle durch die Suche abgedeckt sind, wobei falsche Treffer bei der anschließenden manuellen Analysearbeit aussortiert werden können.

3.3.1 | Precision

Definition

Precision bezeichnet im Kontext der Evaluation von Korpussuchen die Treffergenauigkeit, die sich bei einer gegebenen Treffermenge aus dem Verhältnis von korrekten Treffern und nicht korrekten Treffern ergibt.

Die Precision wird durch einen Wert angegeben, der dem Quotienten aus den korrekten Treffern und allen durch die Suche ermittelten Treffern entspricht. Hierdurch ist die Precision maximal bzw. im besten Fall eins (alle gefundene Treffer sind dann korrekte Treffer) und minimal bzw. im schlechtesten Fall null (keiner der ermittelten Treffer ist in diesem Fall korrekt). (Hinweise: Runden Sie den Wert für Precision auf zwei Nachkommastellen. Entspricht die ermittelte Precision keinem Wert zwischen null und eins, so muss ein Rechenfehler vorliegen.)

Bei Treffermengen, die manuell überprüft werden können, ist die Suchgenauigkeit in Form eines Precision-Werts relativ leicht zu ermitteln. Man teilt dann einfach die Zahl aller Treffer, die die Korpussuche ausgegeben hat, durch die Menge der Treffer, die nach einer Bereinigung der Ergebnisliste übrig bleibt. Häufig ist eine Bereinigung der Suchergebnisse ohnehin erwünscht, so dass sich die Suchgenauigkeit bei der Bereinigungsarbeit sozusagen von selbst ergibt.

Beispielszenario für die Berechnung von Precision: Gegeben sei eine Treffermenge von 3265 Treffern. Von diesen sind 2963 korrekt und 302 falsche Treffer. Dies lässt sich wie folgt tabellarisch darstellen:

	Trefferanzahl
korrekt	2963
nicht korrekt	302
gesamt	3265

Der Wert für Precision beträgt gemäß diesem Szenario 2963/3265 ≈ 0,91. Von den gefundenen Treffern sind also etwa 91 % korrekt.

Bei Treffermengen, die im Gesamten nicht manuell korrigiert werden können, ist es erforderlich, einen möglichst großen Ausschnitt aus den Trefferdaten (bspw. 100 Treffer bei 1000 Treffern) zu evaluieren und den ermittelten Wert für die Genauigkeit annäherungsweise mitzuliefern.

Arbeitsaufgaben

1. Sie möchten Typen von *-bar*-Adjektiven, die aus Verbstämmen und dem Ableitungssuffix *-bar* gebildet wurden, in einem Korpus finden, indem Sie nach Lemmata suchen, deren Form auf *-bar* endet (CQP-Suchausdruck z. B. [lemma = ".*bar"]).
- Ihre Ergebnisliste ist:
  anwendbar, begehbar, Eisbar, verhandelbar, Barbar, brennbar, kämmbar, denkbar, Nachbar, furchtbar, zerstörbar, bar, offenbar, greifbar
- Ermitteln Sie für dieses Suchergebnis den Wert für Precision und geben Sie somit an, wie genau die Suche hinsichtlich der gefundenen Treffer ist.

- *Zusatzaufgabe:* Überlegen Sie auf der Grundlage der genannten Treffermenge, wie Sie die Suchanfrage präzisieren können, damit sich ein höherer Wert für Precision ergibt.

2. Sie haben bei der Suche nach einem beliebigen linguistischen Phänomen rund 12.000 Treffer erzielt, die Sie nicht alle manuell überprüfen können. Sie ziehen deshalb aus der Gesamtmenge an Treffern eine Stichprobe von 200 Treffern und evaluieren diese auf Korrektheit. Dabei ermitteln Sie 37 Fehler (Fälle, die eigentlich nicht in die Treffermenge gehören).
Ermitteln Sie den Wert für Precision.

3.3.2 | Recall

Definition

Recall bezeichnet im Kontext der Evaluation von Korpussuchen die Treffergenauigkeit, die sich aus dem Verhältnis der durch den Suchvorgang gefundenen positiven Treffer und der durch den Suchvorgang nicht gefundenen Treffer ergibt.

Angegeben wird der Recall durch einen Wert, der dem Quotienten aus allen im Korpus enthaltenen Treffern und den tatsächlich gefundenen Treffern entspricht. Hierdurch ist der Recall maximal bzw. im besten Fall eins (alle findbaren Treffer entsprechen dann der Anzahl der tatsächlich gefundenen Treffer) und minimal bzw. im schlechtesten Fall null (in diesem Fall wurde von allen findbaren Treffern kein einziger gefunden).

Der Recall-Wert kann ideal bei der Evaluation automatischer Analysen ermittelt werden, wenn gegen einen Goldstandard verglichen wird, anhand dessen man die Gesamtzahl der zu vergebenden Kategorien ermitteln kann (s. Kap. 2.5.3.1). Ansonsten besteht das Problem bei der Ermittlung des Recalls immer darin, dass die Anzahl sämtlicher Fälle, auf die eine Suche abzielt, benötigt wird. Diese wird für die Gesamtmenge der dursuchten Korpusdaten nicht verfügbar sein. Somit muss man den Recall-Wert grundsätzlich auf einem Ausschnitt der Korpusdaten ermitteln. Man kann z. B. überprüfen, wie häufig eine gegebene Kategorie innerhalb der Menge an Korpusdaten, in der sie mittels des Suchprozesses 100 Mal identifiziert wurde, tatsächlich auftritt.

Beispielszenario für die Berechnung von Recall: Sie erfassen Sätze in einem Korpus durch die Suche nach finiten Verben (STTS-Tag: V.FIN, wobei der Punkt für ein beliebiges Zeichen steht und somit Voll- Hilfs und Modalverben gemäß dem STTS-Tagset abdeckt). In einem Korpusausschnitt von genau 100 als finite Verben ausgewiesenen Wortformen stellen Sie fest, dass zwei Formen fälschlicherweise als finite Verben analysiert wurden und dass überdies sechs ~~Fälle existieren, in denen~~ Sätze ~~ohne finite Verben gebildet werden~~ (Nominalsätze bzw. elliptische Struk-

turen wie *Großartiges Ereignis.* oder Koordinationsellipsen wie *Mary ist einkaufen gegangen und Paul joggen.*, die Sie als zwei Sätze zählen.

Die zahlenmäßigen Verhältnisse lassen sich wie folgt darstellen:

	Trefferanzahl
gefundene korrekte Treffer	98
nicht gefundene Treffer	6
gesamt	104

Der Wert für Recall beträgt gemäß diesem Szenario 98/104 ≈ 0,94. Innerhalb des untersuchten Korpusausschnitts wurden also durch die verwendete Suche etwa 94 % der relevanten linguistischen Strukturen (im konkreten Fall Sätze) gefunden.

Arbeitsaufgabe

Sie möchten mithilfe einer Wortartensuche finite Verbformen aus einem Korpus extrahieren (CQP-Suchausdruck z. B. [pos = "V.*FIN"]). In einem Korpusausschnitt mit 200 als finite Verben getaggten Wortformen stellen Sie fest, dass sechs Treffer falsche Treffer sind, außerdem wurden acht finite Verben nicht als solche (sondern als infinit oder andere Wortarten) erkannt.
Ermitteln Sie für dieses Suchergebnis den Wert für Recall und geben Sie somit hinsichtlich des überprüften Korpusausschnitts an, wie genau die Suche hinsichtlich der eigentlich zu findenden Elemente im Korpus ist.

3.3.3 | F-score

Definition

Der **F-score** ist ein Maß, das zwischen dem ermittelten Wert für Precision und dem ermittelten Wert für Recall mittelt. Hierdurch lässt sich also die allgemeine Genauigkeit von Korpussuchen darstellen.

Die Mittelung zwischen Precision und Recall wird dadurch erzielt, dass der zweifache Faktor der Werte durch die Summe der Werte geteilt wird:

$$F = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

Hierdurch ist der maximale Wert des F-scores eins (wenn Precision und Recall eins betragen) und die Unterschiede zwischen Precision und Recall

werden gemittelt, ohne dass einem der Werte ein höherer Rang zugewiesen wird.

Anwendungsbeispiel zur Ermittlung des F-scores: Stellen Sie sich vor, eine Suchanfrage liefert fast nur positive Treffer: Von 500 gefundenen Treffern sind vier Treffer falsch. Dies ergibt einen Wert für Precision von 0,99. Innerhalb der Datenmenge, in denen die 500 Treffer gefunden wurden, verbergen sich aber 423 weitere positive Fälle. Dies ergibt einen Wert für Recall von nur 0,54.

$$F = \frac{2 \times (0{,}99 \times 0{,}54)}{0{,}99 + 0{,}54}$$

Der aufgelöste Wert für F beträgt 0,7.

Arbeitsaufgabe

Sie möchten in einem wortartgetaggten Korpus die Fokuspartikeln *auch*, *schon* und *nur* untersuchen und formulieren dazu eine Suche nach den drei Lemmata, gefolgt von Artikelwörtern, Präpositionen, pränominalen Adjektiven, Nomina oder Eigennamen (CQP-Suchausdruck z. B. [lemma = «(auch|schon|nur)«][pos = «(ART|P.*AT|APPR|ADJA|N(E|N)«]). Innerhalb von 100 Treffern ermitteln Sie, dass es sich bei 14 Fällen um keine Fokuspartikeln handelt, weil das gemeinsame Auftreten der beiden gesuchten Token unabhängig voneinander erfolgt und die Lexeme *auch*, *schon* und *nur* somit als Adverbiale zu interpretieren sind. Sie stellen außerdem fest, dass innerhalb der untersuchten Textmenge acht Fälle von *auch*, *schon* und *nur* auftreten, in denen es sich um Fokuspartikeln handelt, die allerdings nicht linksadjazent zur Bezugskonstituente stehen oder vor Pronomina auftreten.
Berechnen Sie für die besagte Suchanfrage innerhalb des analysierten Korpusausschnitts den F-score und geben Sie somit für den untersuchten Korpusausschnitt einen Wert für die allgemeine Suchgenauigkeit an.

4 Praxisteil III: Statistische Auswertung von Korpusdaten

4.1 Vorbereitung von auszuwertenden Daten: Datenexportformate und Konversionsszenarien
4.2 Vorbereitende Überlegungen: Typen von Statistik
4.3 Frequenzauswertungen: Erstellung von Frequenzlisten
4.4 Studientypen: verschiedene Klassifikationsansätze
4.5 Methoden für die kontrastive Analyse mindestens zweier Varietäten
4.6 Methoden für die Analyse einer bestimmten Varietät
4.7 Korrelationen

Die in Kapitel 3 behandelten Suchmöglichkeiten in Korpusdaten befähigen ~~den~~ Anwender dazu, Belege für beliebige linguistische Strukturen zu sammeln, sofern dies die Korpusannotationen hergeben, und somit ggf. auch Häufigkeiten von bestimmten linguistischen Kategorien in gegebenen Korpora zu erheben. Viele linguistische Anliegen und Fragestellungen erfordern aber statistische Auswertungen, die über das Auffinden bestimmter Exemplare im Korpus hinausgehen. Wenn man z. B. die Verteilung bestimmter linguistischer Varianten in einem Korpus visualisieren will oder die statistische Signifikanz einer Abweichung zwischen verschiedenen Datenpopulationen errechnen möchte, benötigt man Verfahren, die über bloße Suchmöglichkeiten hinausgehen, und Werkzeuge, die andere Funktionen besitzen als ein Korpus-Suchprogramm.

Eine fundierte Einführung in die Statistik mit ~~samt den erforderlichen~~ theoretischen Grundlagen erfordert eine eigene Monographie und kann in dem hier zur Verfügung stehenden Rahmen nicht geleistet werden. In den folgenden Kapiteln wird deshalb versucht, ~~lediglich~~ einige Grundprinzipien der Korpusauswertung mit quantitativen Methoden zusammenzutragen. Hierdurch sollen grundlegende Herangehensweisen und Probleme (»Fallen« bei der Auswertung) aufgezeigt werden. Einführungen in statistische Methoden für linguistische Daten sind u. a.:

- Baayen (2008) (eine Sammlung statistischer Verfahren auf Englisch, für Fortgeschrittene);
- Gries (2008) und (2013) (ebenso Zusammenstellungen von statistischen Methoden, Deutsch und Englisch, für Einsteiger und Fortgeschrittene);
- Levshina (2015) (eine Zusammenstellung linguistischer Auswertungsmethoden mit dem frei verfügbaren Statistikprogramm R bzw. der Nutzeroberfläche RStudio, für Einsteiger und Fortgeschrittene);
- Lüdeling/Kytö (2008) und (2009) (eine zweibändige Sammlung korpuslinguistischer Ressourcen, Konzepte und Methoden mit vielen Hinweisen auf statistische Zugänge).

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_4

4.1 | Vorbereitung von auszuwertenden Daten: Datenexportformate und Konversionsszenarien

Möchte man Korpusdaten statistisch auswerten, so ergeben sich in der Praxis zwei grundlegende Möglichkeiten:

1. Man nutzt Korpussuchsysteme, um die relevanten Daten aus dem Korpus zu extrahieren oder innerhalb des Suchwerkzeugs Frequenzen zu bestimmten Kategorien zu ermitteln. In den wenigsten Fällen werden aber diejenigen Auswertungsmöglichkeiten von dem jeweiligen Korpussuchwerkzeug angeboten, die im Einzelfall benötigt werden. Deshalb muss im Regelfall mit weiteren Programmen ~~in den aus der Korpussuche hervorgegangenen Suchergebnissen~~ weitergearbeitet werden.
2. Man verarbeitet das Korpus in seiner Gesamtheit, indem man die Text- oder Transkriptionsdaten mit allen Annotationen in ein statistisches Auswertungsprogramm einliest und auswertet. Dieser Lösungsansatz erspart einem den Schritt der Korpussuche, führt aber dazu, dass der Verarbeitungsprozess komplexer wird.

Gemäß beiden Auswertungsszenarien sind in aller Regel Konversionsprozesse notwendig, weil das Eingabeformat des verwendeten Statistikprogramms nicht dem Format entspricht, in welchem die von dem Suchwerkzeug ausgegebenen Daten vorliegen bzw. in welchem das Korpus vorliegt.

Typische Konversionsszenarien sind die folgenden.

1. Aus der KWIC-Ausgabe die eigentlichen Treffer isolieren: Viele Suchsysteme geben die Korpusbelege zu einer bestimmten Suchanfrage als Treffer im Kontext aus, wobei der Treffer selbst durch bestimmte Zeichen, z. B. Tabulatorzeichen, Spitzklammern, @-Zeichen usw., markiert ist. In der Auswertung kann der Trefferkontext störend sein, wenn lediglich der Treffer selbst statistisch weiterverarbeitet werden soll. Ein mögliches Beispiel ist der Ausschnitt aus einer CQP-Suchausgabe (s. Abb. 4.1).

Abb. 4.1 zeigt das Resultat einer Suche nach direkten Abfolgen von Nomina und infiniten Verben in einem CQP-gespeicherten Korpus, gemäß derer das Nomen unmittelbar nach einem Satzzeichen stehen soll, damit nur nicht-begleitete Nomina gefunden werden. Ziel der Auswertung soll sein, die Nomina, die Verben sowie beide Einheiten gemeinsam nach ihrer Frequenz im Korpus aufzulisten. Die gesamte Beispieldatei finden Sie unter der Webadresse https://bit.ly/2TO2juR. Das Ziel, dass bis auf die beiden relevanten Trefferwörter sämtliche Informationen gelöscht werden, lässt sich mit jedem Texteditor erreichen, der reguläre Ausdrücke interpretieren kann, z. B. Notepad + + (https://notepad-plus-plus.org/; s. Arbeitsaufgabe 1 unten für eine entsprechende Konversion der Daten).

Abb. 4.1: Screenshot einer CQP-Trefferausgabe

```
1.  15041:  " Hand anlegen " können <. Apfelsaft pressen> und Bestimmen von Baumen sind
2.  15662:  geschachtelt " umschalten ( anschl <. Aktualisieren drücken> ) . Sie können
3.  208442:  zum Ufer aufsetzen ? Aussteigen <? Schuhe ausziehen> und mit dem Flugzeug
4.  278711:  Ausnahmen von der generellen Danksagung <. Dank sagen> möchte ich zu aller
5.  294662:  , auf die Schinkenscheiben streichen <. Filets aufrollen> und zustecken od
```

```
'34172455','brachte','bringen','VVFIN','34172454','Frühstück','NULL','NULL','Frühstück','NN'
'34172470','wartete','warten','VVFIN','34172473','Weilchen','NULL','NULL','Weilchen','NN'
'34172476','sah','sehen','VVFIN','34172484','Frau','NULL','NULL','Frau','NN'
'34172500','beobachtete','beobachten','VVFIN','34172492','ihn','NULL','CM','er','PPER'
'34172529','gesehen','sehen','VVPP','34172522','den','NULL','CM','d','PRELS'
```

Abb. 4.2: Screenshot einer Trefferausgabe des WEKA-Exporters im ANNIS-Such-interface

2. Aus leerzeichenseparierten Daten ein Spaltenformat erzeugen: Ähnlich einfach lassen sich Tabellenformate erzeugen, die z. B. relevant für die Weiterverarbeitung mit Tabellenkalkulationsprogrammen, der Nutzeroberfläche RStudio für statistische Auswertungen oder anderen Programmen sein können. Vergleichen Sie für ein Beispiel den folgenden Datenauszug, der auf einem Datenexport aus der Suchmaschine ANNIS basiert (s. Abb. 4.2):

Die Daten in Abb. 4.2 resultieren aus seiner Suchanfrage nach Verben und ihren direkten Objekten (Nomen oder Pronomen) in einem Korpus mit syntaktischer Annotation. Mittels des Exporters (»WEKA-Exporter«, s. Kap. 3.1.2.28) wurden Treffer-Token mit Lemma- und Wortartinformationen aus dem Korpus herausgeschrieben. Ein Auswertungsszenario könnte sein, dass die WEKA-Ausgabe in eine CSV-Tabelle mit der Kopfzeile

Verb: Form	Verb: Lemma	Verb: Typ	Obj: Form	Obj: Lemma	Obj: Wortart

umgewandelt werden soll, die sich dann mit einem Statistikprogramm weiterverarbeiten lässt. Ziel ist es also, die Zeichenkette ›,‹ in den ausgegebenen Daten durch Tabulatorzeichen zu ersetzen. Hierfür existiert auch ein regulärer Ausdruck. Beachten Sie die unten stehende Arbeitsaufgabe 2. für eine entsprechende Datenkonversion.

3. Verkettung getrennt stehender Werte: Ein anderes Auswertungsszenario für die in der WEKA-Ausgabe exportierten Daten könnte sein, eine Liste bzw. einspaltige Tabelle mit verketteten Verb- und Objekt-Lemmata zu erstellen, die dann nach der Frequenz von Verb-Objekt-Paaren weiter ausgewertet wird. Hierzu lässt sich die im vorigen Konversionsszenario (s. u. Arbeitsaufgabe 2) erstellte Tabelle weiterverarbeiten, siehe Arbeitsaufgabe 3.

Arbeitsaufgaben

1. Bearbeiten Sie die unter der Webadresse https://bit.ly/2TO2juR verfügbaren Exportdaten so, dass nur die beiden relevanten Wörter innerhalb der Spitzklammern übrig bleiben. Gehen Sie dabei folgendermaßen vor:
   - Laden Sie die angegebene Datei auf Ihren Computer.
   - Öffnen Sie die Datei in einem Texteditor, der beim Suchen und Ersetzen reguläre Ausdrücke unterstützt, z. B. Notepad + + (https://notepad-plus-plus.org/).
   - Öffnen Sie die Funktion »Replace« (bzw. »Ersetzen«; in Notepad + + zu erreichen mit STRG-f oder im Menü unter »Search« > »Replace«).

- Geben Sie in der Suche an, dass Sie reguläre Ausdrücke verwenden (»Regular expression«).
- Geben Sie in das Suchfenster ».* <« (ohne Anführungszeichen) ein. Geben Sie nichts in das Ersetzen-Fenster ein. Sie ersetzen somit sämtliche Zeichen einer Zeile vor der öffnenden Spitzklammer mit nichts, d. h. Sie löschen den String.
- Setzen Sie den Cursor an die erste Position im Dokument und klicken Sie »Replace All«. (Wenn der Cursor irgendwo im Dokument steht, können Sie auch »Replace in All Opened Documents« betätigen.)
- Der linke Trefferkontext bis zur öffnenden Spitzklammer wurde entfernt. Entfernen Sie nun den rechten Kontext, indem Sie das Ersetzen-Prozedere mit der Eingabe »> .*« wiederholen.
- Nun sind lediglich ein Satzzeichen und ein Leerzeichen vor dem relevanten Suchtreffer vorhanden. Diese kann man mit dem Suchbefehl »\n.« löschen (das »\n« bezeichnet einen Zeilenumbruch bzw. -anfang, der Punkt ein beliebiges Zeichen und das Leerzeichen wird wörtlich aufgefasst). Was übrig bleibt, ist die für die Auswertung relevante Zeichenkette.

2. Bearbeiten Sie die unter der Webadresse https://bit.ly/2HMjx4I verfügbaren Exportdaten so, dass eine sechsspaltige Tabelle mit der Kopfzeile wie oben gezeigt entsteht und die einzelnen Werte der Exportdatei korrekt zugeordnet sind. Gehen Sie dabei folgendermaßen vor:
- Gehen Sie bis zur Eingabe des Suchstrings vor wie in Aufgabe 1.
- Geben Sie im Suchfeld den Ausdruck "','" (einfaches Anführungszeichen, Komma, einfaches Anführungszeichen) ein. Geben Sie im Feld für die Ersetzung »\t« ein (die Kombination steht für einen Tabulator-Abstand).
- Kopieren Sie den Inhalt in ein geöffnetes Tabellenblatt einer LibreOffice-Calc-Datei oder in eine Microsoft-Excel-Datei (Zelle A1).
- Löschen Sie die Spalten A, E, G und H (die für die Vorgabe irrelevanten Spalten).
- Fügen Sie ganz oben im Dokument eine Zeile hinzu.
- Überschreiben Sie die Spalten mit den oben angegebenen Werten.
- Speichern Sie die Datei als Trennzeichen-getrennte Datei unter dem Namen »Konvertieren_2_konvertiert.csv«.

3. Bearbeiten Sie die in Aufgabe 2 erstellte Datei weiter, indem Sie die Werte der Spalten »Verb: Lemma« und »Obj: Lemma« mit einem Unterstrich verbunden zusammenführen und die Kopfzeile löschen. Gehen Sie hierzu wie folgt vor:
- Öffnen Sie die Datei, die Sie bei der Bearbeitung von Aufgabe 2 gespeichert haben. Die Datei können Sie auch unter der Webadresse https://bit.ly/2FpKqJS beziehen.
- Schreiben Sie in die Zelle G2 den Befehl »=VERKETTEN(B2;«_«;E2)« (ohne Anführungszeichen) und betätigen Sie Enter.
- Markieren Sie die Zelle G2 und klicken Sie doppelt auf das Plus rechts unten in der markierten Zelle. Der Befehl wird bis in die letzte gefüllte Zeile kopiert.

- Kopieren Sie den Text von Zelle G2 bis G539 in Zelle H1 bis H540, ohne dass die Formeln mitkopiert werden (»Inhalte einfügen...« > »Werte« bzw. »Formeln« nicht anklicken).
- Löschen Sie alle Spalten bis auf Spalte H und speichern Sie das Ergebnis unter dem Namen »Konvertieren_3_konvertiert.csv«.
  Hinweis 1: Sie können die Konversion auch mit einem Texteditor wie Notepad + + durchführen, indem Sie die zwei relevanten Spalten dorthin kopieren und den Tabulatorabstand (»\t«) durch einen Unterstrich ersetzen.
  Hinweis 2: In Kap. 4.3 wird behandelt, wie man ~~die~~ Daten wie die in dieser Aufgabe erstellten nach Frequenzen auswertet. So kann man z. B. analysieren, dass die Verbindung »bringen_Frühstück« mit genau acht Vorkommen die häufigste Verb-Objekt-Verbindung in den zugrunde liegenden Korpusdaten ist (es handelt sich um ein kleines, sehr spezifisches Korpus mit Kafka-Texten).

4.2 | Vorbereitende Überlegungen: Typen von Statistik

Definition

Deskriptive (beschreibende) Statistik: Korpusbasierte Ansätze des deskriptiven Typs haben zum Ziel, die Korpusdaten bezogen auf bestimmte Phänomenbereiche hin abzubilden. Verfahren, die Häufigkeiten bestimmter Kategorien ermitteln und darstellen oder die die Verteilung bestimmter Kategorien in Korpora ermitteln und darstellen, gehören zu diesem Typus. Deskriptive Statistik hat auch immer einen zusammenfassenden Charakter, indem sie bestimmte Merkmale eines Korpus oder mehrerer Korpora übersichtlich zusammenträgt.
Analytische (erklärende, auch ›inferentielle‹, schließende) Statistik: Hierbei geht man einen (oder mehrere) Schritt(e) weiter, weil man nicht nur aufzeigen möchte, wie die Korpusdaten aussehen, sondern mittels statistischer Verfahren Schlussfolgerungen ~~erzielen~~ möchte. Diese können sich z. B. auf die Beziehung der untersuchten Korpusdaten und die Welt richten (Frage: »Beschreiben die Korpusdaten nur ein individuelles Korpus oder darf man die im Korpus sichtbaren Tendenzen verallgemeinern und somit auf die eigentliche Welt beziehen?«, s. Kap. 4.5.4). Sie können sich auch auf Zusammenhänge innerhalb von Korpora beziehen, indem z. B. nachgewiesen wird, dass ein beobachtetes gemeinsames Auftreten von Wörtern (oder anderen Strukturen) auf einer tatsächlichen Anziehung dieser Elemente beruht und nicht auf Zufall ~~oder auf Konsequenzen, die auf den Korpusdaten selbst beruhen~~
(s. Kap. 4.6.1). Somit haben analytische statistische Verfahren einen schlussfolgernden, verallgemeinernden Charakter.

Ziel fast jeder korpuslinguistischen Auswertung ist die Untersuchung der durch Korpora bereitgestellten Sprachdaten mithilfe von statistischen Methoden. Ausnahmen sind das Auffinden von Belegen zu einem gewissen linguistischen Phänomen oder das Zusammenstellen von Listen zu bestimmten linguistischen Mustern.

Es ist wichtig zu beachten, dass man mit Statistik grundlegend unterschiedliche Ziele verfolgen kann. Im Wesentlichen gibt es die zwei im Definitionskasten genannten Typen von Statistik. Zu beiden Typen gehören wiederum viele konkrete Auswertungsansätze.

Häufig werden statistische Verfahren so miteinander gekoppelt, dass zunächst deskriptiv und anschließend analytisch gearbeitet wird. Die folgenden Kapitel behandeln dementsprechend zunächst deskriptive, dann analytische Methoden.

4.3 | Frequenzauswertungen: Erstellung von Frequenzlisten

In fast allen Korpusuntersuchungen werden Frequenzen von Kategorien ermittelt. Diese können dann unmittelbar oder mittelbar (s. Kap. 4.5.1) miteinander verglichen werden, miteinander korreliert werden, anhand von Frequenzen können Assoziationsstärken ermittelt und somit Kollokationen oder Gebrauchstendenzen nachgewiesen werden (und vieles mehr). In den folgenden Kapiteln wird behandelt, wie man Frequenzinformationen aus Korpora herauszieht und wie man mit diesen Informationen dann weiterarbeiten kann.

Frequenzlisten wie z. B. eine Liste der 20 häufigsten Wörter (Lemmata) oder die 20 häufigsten Abfolgen von Präposition und Artikel in einem Korpus basieren zuallererst auf Suchprozessen, die in Kapitel 3.1 und 3.2 behandelt wurden. Hier wurden auch wesentliche Möglichkeiten des Datenexports und der Quantifizierung der Suchergebnisse dargestellt.

Analyseszenario: Wenn z. B. alle Eigennamen in einem Korpus der Häufigkeit nach aufgelistet werden sollen, so ist dies mit den meisten der in Kapitel 3.1 und 3.2 besprochenen Korpussuchsystemen unmittelbar machbar: Es gilt, eine Suche nach allen Eigennamen im Korpus zu definieren und eine Ausgabe der Treffer nach Lemmata und deren Häufigkeit im Korpus zu erzielen. Befolgen Sie die Arbeitsschritte in der Anleitung, um exemplarisch eine Frequenzauswertung von Eigennamen im ANNIS-Suchinterface vorzunehmen.

Anleitung

- Wählen Sie im Suchinterface https://hu.berlin/annis-intro das Korpus »Fuerstinnenkorrespondenz1.k.aus und stellen Sie die folgende Suchanfrage, die Eigennamen findet und gleichzeitig auf die zugehörigen Lemmata zugreift:
  pos = «NE« _ = _ lemma
- Wählen Sie im Interface die Funktion »More« > »Frequency Analysis«.

- Löschen Sie die erste Suchvariable »1 – pos« aus der Übersicht, indem sie die Zeile markieren und »Delete selected row(s)« wählen.
- Wählen Sie nun »Perform frequency analysis«.
- Sie erhalten eine graphische und eine tabellarische Übersicht zu allen Eigennamen im Korpus, sortiert nach Lemma (zur Sortierung nach Wortform durchlaufen Sie die Prozedur mit der Variable »tok« statt »lemma«.

Ist die interne Frequenzauswertung in einem bestimmten Suchwerkzeug nicht möglich, so müssen die Suchtreffer einzeln herausgeschrieben werden und die Frequenzauswertung muss extern erfolgen. In dem Fall gehen Sie so vor, wie im Folgenden beschrieben.

~~Oft ist es mit den Suchsystemen, die auf die Korpusdaten mit allen Korpusannotationen zugreifen können, nicht möglich, statistische Auswertungen vorzunehmen, so dass man auf weitere Korpuswerkzeuge angewiesen ist. Häufig besteht das Problem, dass an die Frequenzaufstellung bestimmte Anforderungen gestellt werden, wie im erweiterten Analyseszenario der Fall.~~

Erweitertes Analyseszenario: Aus dem im ANNIS-Suchinterface (https://hu.berlin/annis-intro) durchsuchbaren Korpus »Fuerstinnenkorrespondenz1.1« sollen zu jedem Text die Anzahl der dort enthaltenen Eigennamen angegeben werden. Es soll also eine Frequenzliste entstehen, die die einzelnen Dokumentnamen nach der Anzahl der in ihnen vorkommenden Eigennamen auflistet. (Die Dokumentnamen sind im gegebenen Fall als Korpusmetadaten zur Variable »doc« für »Dokument« abgelegt.)

Das angegebene Ziel ist ~~prinzipiell~~ zu erreichen, indem man per Metadatenabfrage pro Text-Metadatum nach allen Eigennamen sucht und die jeweiligen Frequenzen in eine gesonderte Tabelle schreibt. (Die Suchanfrage für den ersten Text im Korpus wäre: pos = «NE« & meta::doc = «AD_JE2_1677_08_14«; die Suche erzielt vier Treffer.) Da das Korpus 600 Texte beinhaltet, wäre dies jedoch ein erheblicher Aufwand. Eine gangbare Lösung ist, alle Suchtreffer zu Eigennamen mitsamt den dazugehörigen Dokument-Werten zu exportieren und diese Ergebnisse anschließend zu quantifizieren. Um dies zu bewerkstelligen, gehen Sie gemäß den in der Anleitung beschriebenen Schritten vor.

Anleitung

- Wählen Sie im Suchinterface https://hu.berlin/annis-intro das Korpus »Fuerstinnenkorrespondenz1.1« aus und stellen Sie die folgende Suchanfrage, die Eigennamen findet:
  pos = «NE«
- Wählen Sie im Interface die Funktion »More« > »Export«.
- Lassen Sie den »CSVExporter« angewählt, stellen Sie die Ausgabekontexte auf null und geben Sie im Feld »Paramters« »metakeys = doc« (ohne Anführungszeichen) ein. Hierdurch wird bei jedem Treffer der Dokumentname mit exportiert.

- Klicken Sie »Perform Export« und laden Sie die erzeugte Datei herunter oder öffnen Sie sie gleich in einem Texteditor.
- Zur Vorbereitung der Daten für die weitere Verarbeitung ist das Ziel, lediglich noch eine einspaltige Tabelle bzw. Liste der Dokumentnamen zu behalten. Dies erzielt man dadurch, dass man mithilfe eines regulären Ausdrucks pro Zeile die gesamte Zeichenkette bis zum letzten Tabulatorzeichen löscht bzw. durch nichts ersetzt. Sie können auch wie folgt vorgehen:
  - Kopieren Sie die von ANNIS exportierten Daten nach LibreOffice (oder OpenOffice) Calc oder Microsoft Excel oder öffnen Sie die Datei so, dass eine vierspaltige Tabelle angezeigt wird.
  - Kopieren Sie nur die rechte Spalte der Dokumentnamen zurück in die Textdatei oder löschen Sie die drei ersten Spalten und speichern das Ergebnis als .txt-Datei ab.
  - Achten Sie darauf, dass die Kopfzeile des ANNIS-Exports gelöscht ist bzw. dass die oberste Zeile der erste Dokumentname ist.

Erstellung von tabellarischen Listen mit AntConc: In Kapitel 3.1.1 zur Korpussuche mit AntConc wurde das Programm AntConc (http://www.laurenceanthony.net/software/antconc/ Anthony 2018) bereits als ein leicht zu handhabendes Werkzeug für einfache Suchbelange vorgestellt. Auch bei der Auswertung von Korpusdaten bietet AntConc grundlegende Möglichkeiten, die wenig technischen Aufwand erfordern.

Im gegebenen Fall soll AntConc die in der aktuellen Datei befindlichen Werte nach Frequenz ordnen [– dann ist das Analyseziel erreicht]. Wann immer mit AntConc sämtliche Elemente aus einer (oder mehreren) tokenisierten Textdatei(en) nach Frequenz geordnet werden sollen, können Sie wie folgt vorgehen:

Anleitung

- Öffnen Sie AntConc und lesen Sie die zu bearbeitende Datei(en) ein.
- Wählen Sie oben im Reiter die Funktion »Word List«.
- Stellen Sie sicher, dass AntConcs Tokendefinition zu den zu zählenden Daten passt: Antconc segmentiert die eingelesenen Daten standardmäßig an allen Zeichen, die keine Buchstaben sind. Das ist im vorliegenden Fall ein Problem, weil die Dokumentnamen des analysierten Korpus Zahlen und Unterstriche enthalten. Wählen Sie also unter »Global Settings« > »Token Definition« die Optionen »Number« und »Punctuation« hinzu oder wählen Sie »Use Following Definition« an und geben Sie der Liste an Zeichen alle Zahlen und den Unterstrich hinzu. Drücken Sie anschließend »Apply«.
- Wählen Sie nun unter der Auswahl »Word List« »Start«. Sie erhalten die gewünschte Liste an Frequenzen pro Textdokumentnamen.
- Wählen Sie »File« > »Save Output« und speichern Sie das Ergebnis.

4.4 | Studientypen: verschiedene Klassifikationsansätze

Für die korpuslinguistische Methodologie lassen sich verschiedene Studientypen ausmachen. In Kapitel 4.5 und 4.6 werden die verschiedenen zu besprechenden Methoden nach dem grundlegenden Forschungsziel sortiert, und zwar danach, ob die korpuslinguistische Analyse die vergleichende Untersuchung zweier Varietäten (Sprachen) oder die Untersuchung einer Varietät oder verallgemeinerbarer grammatischer Gesetzmäßigkeiten anstrebt, die keiner Vergleichsvarietät bedürfen.

Geläufig sind vor allem zwei Ansätze:

1. **Querschnittstudie vs. Längsschnittstudie:** Querschnittstudien untersuchen linguistische Merkmale, z. B. das Kasussystem, zu einem bestimmten Zeitpunkt, z. B. im aktuellen Deutsch oder um 1600 herum. Auch der Vergleich zweier Varietäten zu einem gegebenen Zeitpunkt entspricht einer Querschnittstudie. Längsschnittstudien dagegen untersuchen linguistische Merkmale, z. B. das Kasussystem des Deutschen, im zeitlichen Verlauf, indem dieselbe linguistische Analyse zu Sprachdaten aus verschiedenen historischen Zeiträumen oder -punkten wiederholt wird.
2. **Studientyp A, B und C nach Biber/Jones 2009:** Diese Unterscheidung beruht auf dem Typ der im Fokus stehenden Einheiten. Typ-A-Studien haben ein linguistisches Phänomen im Blickpunkt, zu dem verschiedene Varianten gehören können (Biber/Jones 2009 nennen das Beispiel des Relativsatzes, der verschiedene Erscheinungsformen, z. B. verschiedene Anschlusswörter, haben kann). In Typ-A-Studien geht es um die Ergründung und Erklärung dieses Phänomens durch dessen Analyse innerhalb einer bestimmten Datenpopulation, eines bestimmten Korpus. Typ-B-Studien zielen auf den quantitativen Vergleich von linguistischen Merkmalen in verschiedenen Varietäten ab, um so die Varietäten beschreiben zu können. Hierzu können mehrere Korpora oder ein in Subkorpora segmentierbares Korpus dienen (Subkorpora sind durch Metadaten filterbare Unterkorpora zu einem Korpus). Wichtig ist den Autoren, dass jeder einzelne Text in den verglichenen Datenpopulationen in die statistische Auswertung eingeht, um die Analyse statistisch zu validieren (s. hierzu vor allem Kap. 4.5.3 und 4.5.4). Typ-C-Studien verfolgen dasselbe Ziel wie Typ-B-Studien, doch ihnen fehlt die Größe des Texts in der statistischen Analyse, so dass Vergleiche über Varietäten hinweg nur anhand von Mittelwerten erfolgen und statistisch nicht validiert werden können (s. hier ebenso Kap. 4.5.3 und 4.5.4).

Im weitesten Sinn entsprechen die Studientypen B und C nach Biber/Jones (2009) den im Kapitel 4.5 besprochenen Verfahren und der Studientyp A den in Kapitel 4.6 behandelten Methoden.

4.5 | Methoden für die kontrastive Analyse mindestens zweier Varietäten

Definition

> Die **Korpusgröße** ist bei dem Vergleich von zwei oder mehr Korpora ein entscheidender Faktor. Auch wenn sie standardmäßig in der Tokenzahl pro Korpus angegeben wird, kann sie nicht einheitlich definiert werden, weil sie phänomenbezogen ist. Dies wird in Kap. 4.5.1–4.5.4 ausführlich dargelegt.

Auf der Grundlage der bisher behandelten Methoden zur Ermittlung von Frequenzen aus Korpusdaten werden wir nun behandeln, wie man solche Auswertungen zwischen verschiedenen Korpora vergleicht. Hierbei muss vor allem berücksichtigt werden, dass zwei Korpora nie gleich groß sind. Selbst wenn die Korpusgröße, angegeben in der Tokenzahl oder Anzahl der Textwörter, annähernd gleich oder gar identisch sein sollte, ist nicht zu erwarten, dass die Gesamtheit des ausgewerteten Phänomens dieselbe ist.

Anwendungsbeispiel: Stellen Sie sich vor, Sie haben zwei Korpora mit jeweils exakt 100.000 Token aggregiert. Das eine Korpus repräsentiert ein (medial und konzeptionell) gesprochenes, das andere ein (medial und konzeptionell) geschriebenes Register. Es ist kaum eine linguistische Fragestellung denkbar, die es rechtfertigte, absolute Zahlen der Vorkommen eines bestimmten linguistischen Phänomens direkt miteinander zu vergleichen. Jede morphologische, syntaktische oder textlinguistische Fragestellung bedingt, dass die ~~Vergleichbarkeit~~ der Tokenzahl irrelevant wird. Stellen Sie sich vor, Sie wollen vergleichen, wie viele Nomina in den jeweiligen Korpora komplex sind: Es wird nicht genügen, die jeweilige Gesamtzahl der komplexen Nomina pro Korpus gegeneinanderzustellen. Denn es muss berücksichtigt werden, dass die Anzahl der Nomina selbst unterschiedlich sein kann (und auch sein wird). Somit ist irrelevant, dass die Tokenzahl der Korpora ~~übereinstimmt~~. Was hinsichtlich des Vergleichs komplexer Nomina übereinstimmen muss, ist die Gesamtzahl der Nomina. Diese Vergleichsproblematik wird genauer in Kapitel 4.5.1 besprochen. Die Aussagekraft verglichener Häufigkeiten zwischen zwei Korpora behandeln die Kapitel 4.5.3 und 4.5.4.

4.5.1 | Zählung und Normalisierung

Normalisierung bezeichnet einen Rechenprozess, bei dem absolute Häufigkeiten in relative Häufigkeiten umgerechnet werden, um die Werte mit anderen Werten vergleichbar zu machen. Entscheidend dabei ist die Normalisierungsgröße, die sich aus der Gesamtzahl für die Zählung relevanter Ereignisse ergibt.

Definition

Das Problem bei dem Vergleich von Häufigkeiten bestimmter grammatischer Phänomene in verschiedenen Datenpopulationen ist, dass die Populationen praktisch nie gleich groß sind. Aus diesem Grund sind auch absolute Werte, die die Auftretenshäufigkeit bestimmter Phänomene darstellen, nicht miteinander vergleichbar. Betrachten Sie hierzu das folgende Szenario: Sie wollen die Häufigkeit von Nomina in verschiedenen Korpora miteinander vergleichen. Dies ist sinnvoll, wenn Sie annehmen, dass Sie damit eine allgemeingültige Aussage zur Häufigkeit von Nomina in den entsprechenden Varietäten ableiten können.

Untersuchungsszenario zur Illustration: Für eine Studie sollen zwei STTS-getaggte Korpora nach dem Wortarten-Tag »NN« (normales Nomen) ausgewertet werden. Das eine Korpus besteht aus transkribierten Gesprächen. Die Korpusgröße wird mit 11.199 Token bemessen. Das andere Korpus besteht aus Zeitungstexten der »Frankfurter Allgemeinen Zeitung«; die Tokenzahl beträgt 10.832. Dies trifft tatsächlich auf die Korpora »BeMaTaC_L1_3.0« (als Gesprächskorpus; https://hu.berlin/bematac/) und »NoSta-D-TueBaDZ« (als Zeitungskorpus; https://hu.berlin/nostad/) zu, die im ANNIS-Suchinterface der Humboldt-Universität zu Berlin frei verfügbar sind (https://hu.berlin/annis/). Die Auswertung der Korpussuchen nach dem Wortartentag »NN« lässt sich wie folgt darstellen.

Korpusname	Häufigkeit »NN«
BeMaTaC_L1_3.0	1056
NoSta-D-TueBaDZ	1878

Wenn man berücksichtigt, dass die Korpusgrößen, angegeben in Token, relativ vergleichbar sind, könnte man schlussfolgern, dass Nomina in den Zeitungsdaten häufiger sind als in den Gesprächsdaten. Es ist jedoch nicht möglich, ein genaues Verhältnis auszudrücken, weil sich die absoluten Häufigkeiten der Nomina auf verschieden große Datenpopulationen beziehen. Um vergleichbare Zahlen zu erhalten, muss man die absoluten Häufigkeiten an einer Bezugsgröße messen. Die gröbste – und in den meisten Fällen falsche – Bezugsgröße ist die Tokenzahl für das jeweilige Korpus. In dem konkreten Fall darf man die Tokenzahl nicht als Normalisierungsgröße nehmen, weil in dem Gesprächskorpus viele Token gar keine Wortart zugewiesen haben, wenn es sich nämlich nicht um sprach-

liche Einheiten, sondern z. B. um Pausen handelt. Anders ausgedrückt, besitzen die zwei verglichenen Korpora eine andere Definition der Kategorie »Token«. Als alternative Messgröße kann und sollte man als Normalisierungsgröße die Gesamtzahl der in den jeweiligen Korpora vergebenen Wortarten nehmen. Dementsprechend ergibt sich folgende Auswertung.

Korpusname	Häufigkeit »NN«	Gesamtzahl STTS-Tags	normalisierter Wert pro 100 STTS-Tags
BeMaTaC_L1_3.0	1056	8835	12,0
NoSta-D-TueBaDZ	1878	10830	17,3

Die gerundeten Werte sind nun unmittelbar miteinander vergleichbar, denn sie sagen aus, wie häufig der STTS-Wert »NN« pro 100 STTS-Werte jeweils ist. Bildet man aus diesen Werten den Quotienten (0,694), so besagt dieser, dass die Häufigkeit von »NN« in dem Gesprächskorpus gerade einmal knapp 70 % von dem Vorkommen im Zeitungskorpus beträgt. Wie man ermittelt, ob man dieses in den Korpora gemessene Verhältnis verallgemeinern und somit auf die Varietäten, die die jeweiligen Korpora repräsentieren, beziehen darf, wird in Kapitel 4.5.4 erläutert.

Hinweis zum Aufführen absoluter und normalisierter Werte: Führen Sie stets sowohl normalisierte Werte und absolute Werte an, denn während die normalisierten Werte notwendig für die unmittelbare Vergleichbarkeit sind, benötigt man zusätzlich absolute Zahlen, um bezüglich der absoluten Häufigkeit nicht getäuscht zu werden. Stellen Sie sich dazu folgende Szenarien vor: Sie werten zwei größere Korpora auf den Gebrauch von Modalpartikeln aus (Wörter wie *wohl*, *doch*, *ja*, *halt* usw. in einer entsprechenden syntaktischen Verwendung). Sie normalisieren dabei das Auftreten der einzelnen Lexeme an der Gesamtzahl der Modalpartikeln im jeweiligen Korpus. Hierdurch wäre z. B. die Aussage denkbar, dass die Modalpartikel *ja* im ersten Korpus die häufigste Modalpartikel ist und gerundet 63 Mal pro einhundert Modalpartikeln auftritt. In einem extremen Fall könnte das erste Korpus insgesamt nur acht Modalpartikeln aufweisen, wovon fünf die Form *ja* haben. Dann ist der Normalisierungswert zwar rechnerisch korrekt, doch die Tatsache, dass das linguistische Phänomen an sich in den untersuchten Daten sehr selten ist, geht durch die Normalisierung verloren. Die Nennung absoluter Zahlen (der Gesamtvorkommen) verhindert dieses verzerrte Bild.

Arbeitsaufgaben

1. Bestimmen Sie anhand der vorangegangenen Informationen und Ihrem linguistischen Wissen die Normalisierungsgröße in den folgenden Vergleichsszenarien: Es werden jeweils zwei (unterschiedlich große) Korpora auf die genannte linguistische Kategorie hin verglichen. An welcher Normalisierungsgröße müssen Sie die absoluten

Werte bzw. ausgezählten Mengen messen, um vergleichbare Zahlen zu erhalten?
a) Nebensätze
b) Verb-erst-Sätze
c) Relativsätze
d) Nominalkomposita
e) Schwa-Tilgungen
f) Wortabbrüche
g) kausale Diskursrelationen
h) Perfektsätze
i) In dialogischen Texten: Unterbrechungen

2. Rechnen Sie die absoluten Werte in vergleichbare Werte um: Gegeben sind jeweils absolute Werte zu einer bestimmten linguistischen Kategorie. Die folgende Tabelle beinhaltet mögliche Normalisierungsgrößen.

	Korpus A	Korpus B
Token	888169	1192032
Textwörter	768190	1029398
Nomina	183977	246584
Verben	106852	143102
Vollverben	68416	92004
Sätze (Haupt- und Nebensätze)	72386	96879
Nebensätze	19888	26617

Entscheiden Sie, welche Normalisierungsgröße jeweils am geeignetsten ist und rechnen Sie den entsprechenden normalisierten Wert pro 100 Einheiten aus. Runden Sie das Ergebnis auf eine Nachkommastelle.

Abstand muss vergrößern

a)

	Korpus A	Korpus B
absoluter Wert: Wörter mit mindestens einem <e> oder <E>	501001	662326

b)

	Korpus A	Korpus B
absoluter Wert: Fragesätze	736	890

c)

	Korpus A	Korpus B
absoluter Wert: Bewegungsverben	2549	1931

d)

	Korpus A	Korpus B
absoluter Wert: Subjunktion *obwohl*	147	202

e)

	Korpus A	Korpus B
absoluter Wert: Attributsätze	8634	11208

f)

	Korpus A	Korpus B
absoluter Wert: modale Adverbiale	34193	46577

3. Berechnen Sie im CQP-Suchinterface (https://hu.berlin/cqp; Login: CQP_Demo, Passwort: TestSuchen) für das Korpus »Akademisches Deutsch« die relativen Häufigkeiten für das Auftreten subordinierender Konjunktionen in den Subkorpora »Chemie« (Metadatenwert: »chemie«), »Physik« (Metadatenwert: »psysik«) und »Medizin« (Metadatenwert: »medizin«). Beantworten Sie die folgende Frage: Sind die Vorkommen in den drei Subkorpora eher ähnlich häufig oder verschieden häufig? Bitte beachten Sie zur Korpussuche mit Metadaten die Informationen in Kapitel 3.1.2.27 (insbesondere Szenario 2).

4.5.2 | Vergleichsdarstellungen und -software

Für den Vergleich verschiedener Datenmengen bieten sich unterschiedliche Möglichkeiten der Darstellung an. Es existieren verschiedene Programme, die bei der statistischen und grafischen Auswertung behilflich sind.

4.5.2.1 | Darstellungsmöglichkeiten

Verfügt man über Frequenzdaten zum Vergleich verschiedener Datenpopulationen, wie sie in Kapitel 4.5.1 exemplarisch gezeigt wurden und in der Arbeitsaufgabe zu erstellen waren, so stellt sich die Frage nach der Darstellung dieser Daten. Grundlegende Möglichkeiten sind die der tabellarischen Darstellung oder der grafischen Darstellung in Form von Balken- bzw. Säulendiagrammen. Vergleichen Sie die verschiedene Darstellungen, basierend auf jeweils demselben Inhalt (s. Tab. 4.1 und Abb. 4.3).

Tabellarische Daten können wie in Tab. 4.1 aufbereitet sein.

Säulendiagramme können genau solche Daten visualisieren (s. Abb. 4.3).

Säulendiagramme wie in Abb. 4.3 lassen sich schnell in Tabellenkalkulationsprogrammen wie LibreOffice (oder OpenOffice) Calc oder Micro-

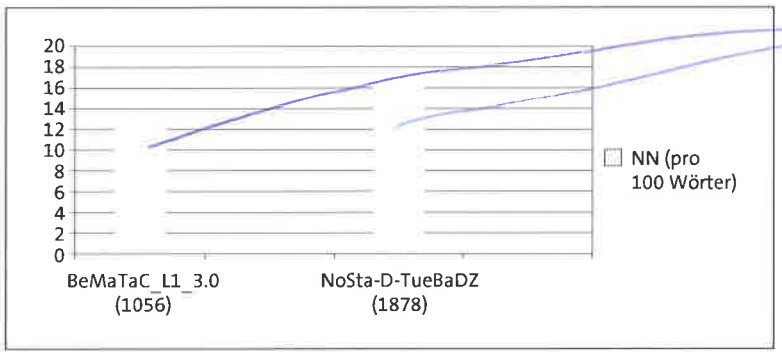


Abb. 4.3: Visualisierung der Werte aus Tab. 4.1

Korpus	Häufigkeit »NN«	normalisiert (pro 100 Wörter)
BeMaTaC_L1_3.0	1056	12,0
NoSta-D-TueBaDZ	1878	17,3

Tab. 4.1: Auswertung von Nomenfrequenzen in zwei Korpora

soft Excel erstellen. Die zugrunde liegenden Daten können z. B. in einer zweispaltigen Tabelle stehen

	NN (pro 100 Wörter)
BeMaTaC_L1_3.0 (1056)	12
NoSta-D-TueBaDZ (1878)	17,3

Durch die Markierung dieser Zellen und dem Befehl zum Einfügen eines Diagramms lässt sich in den genannten Programmen die grafische Darstellung der eingetragenen Daten erstellen. Im Internet findet man bei Bedarf etliche Anleitungen zur Erstellung von Diagrammen mit den allgemein gebräuchlichen Tabellenkalkulationsprogrammen. Vergleichen Sie auch die Arbeitsaufgaben in Kapitel 4.6.1, in denen weitere Hinweise zum Erstellen von Tabellen aus tabellarischen Frequenzdaten stehen.

Andere Diagrammtypen sind das Kreisdiagramm, das Verlaufsdiagramm und das Streudiagramm, die sich für verschiedene Zwecke der Visualisierung von Eigenschaften der analysierten Daten eignen.

Kreisdiagramme eignen sich z. B. dazu, die relativen Anteile bestimmter Kategorien an einer gemeinsamen zugehörigen Variable zu visualisieren (s. in diesem Sinne besonders das Kap. 4.6.1). Das folgende Kreisdiagramm zeigt das jeweilige Verhältnis der STTS-Wortarten Vollverb (VV..), Modalverb (VM..) und Hilfs- bzw. Kopulaverb (VA..) im Korpus »BeMaTaC_L1_3.0« (s. Abb. 4.4).

Mit Verlaufsdiagrammen kann man Entwicklungen über die Zeit oder andere sequenzierbare Größen abbilden. Vergleichen Sie als Beispiel ein

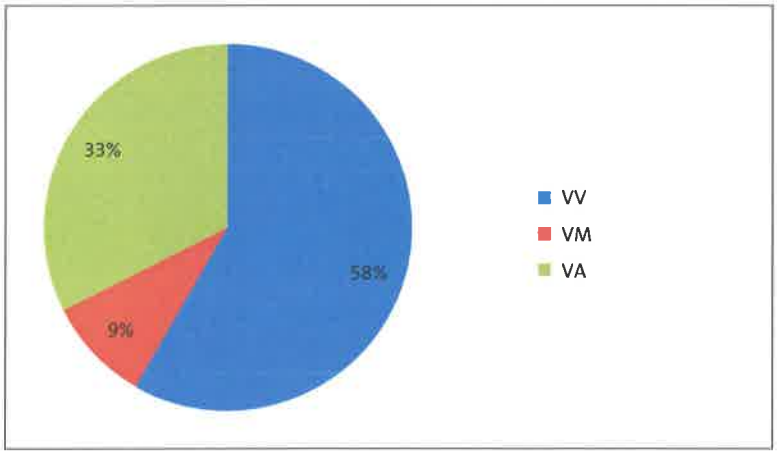


Abb. 4.4: Verteilung der Varianten verschiedener Verbtypen im Korpus »BeMaTaC L1 3.0«

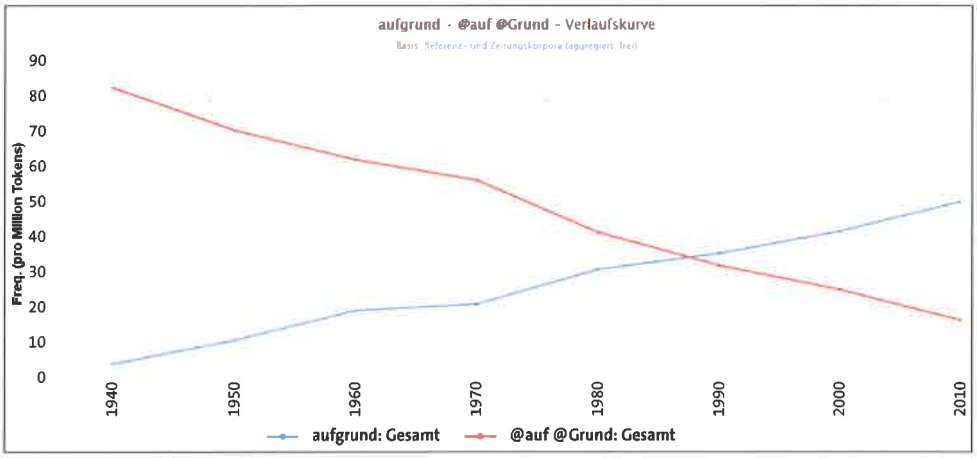


Abb. 4.5: Zeitlicher Verlauf der Varianten *aufgrund* sowie *auf Grund* in den DWDS-Referenz- und Zeitungskorpora (Erstellung unter http://www.dwds.de/r/)

im DWDS-Wortstatistik-Werkzeug (http://www.dwds.de/r/) erzeugtes Verlaufsdiagramm zu den Formen *aufgrund* und *auf Grund* von 1940 bis 2010, gemessen in Zehnjahresintervallen (s. Abb. 4.5).

Gemäß dem angezeigten Verlauf in Abb. 4.5 lässt sich schlussfolgern, dass die synthetisierte Variante um 1990 die analytische Variante an ihrer relativen Gebrauchshäufigkeit überstiegen hat und insgesamt eine langsame Ablösung der analytischen durch die synthetische Variante erfolgt.

Streudiagramme bzw. Punktdiagramme eignen sich dazu, Varianzen innerhalb von Datenmengen aufzuzeigen (s. Kap. 4.5.3) oder Korrelationen zu visualisieren (s. Kap. 4.7). Vergleichen Sie hierzu das Streudiagramm (s. Abb. 4.6).

Das Diagramm in Abb. 4.6 zeigt für jeden Text (= jeder Punkt in der Abbildung) im Korpus »Fuerstinnenkorrespondenz« im ANNIS-Suchinterface (https://hu.berlin/annis-intro) die relative Häufigkeit von Verben und Modaladverbialen. Das Bild legt eine positive Korrelation nahe (s. Kap. 4.7), aber es zeigt sich auch, dass die Streuung der Daten relativ

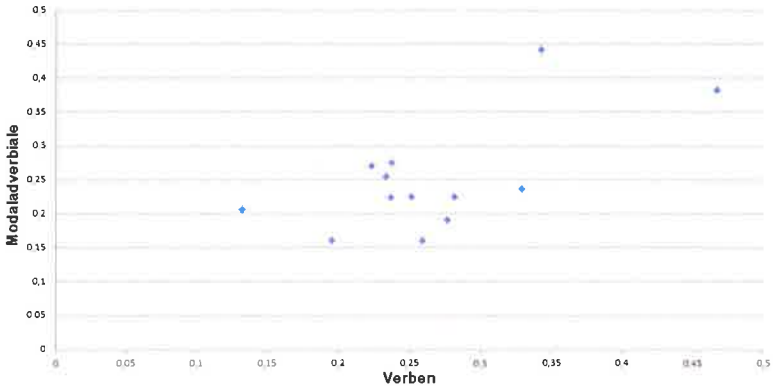


Abb. 4.6: Streudiagramm zur Abhängigkeit von Verben und Modaladverbialen im Korpus »Fürstinnenkorrespondenz«

groß ist, so dass die Stärke der Korrelation nicht als sonderlich hoch eingeschätzt werden kann.

Die gezeigten grundlegenden Visualisierungsmöglichkeiten werden, wo angebracht, in den kommenden Kapiteln zur Datenauswertung eingesetzt.

4.5.2.2 | Vergleich von absoluten und gemittelten Frequenzen mit KoGra-R

Das online zu bedienende Programm KoGra-R (http://kograno.ids-mannheim.de/) ist ein von Mitarbeitern des Instituts für Deutsche Sprache in Mannheim entwickeltes Frequenzauswertungsprogramm, das auf dem Statistikprogramm R (http://www.r-project.org) basiert. Als Eingabe dienen tabellarische Datensätze in drei unterschiedlichen Formatmöglichkeiten (COSMAS-Exportdateien, CSV-Dateien oder die Freitexteingabe einer semikolongetrennten Tabelle. Normalisierte Werte werden aus absoluten Frequenzen und dazugehörigen Normalisierungswerten errechnet. Um eine Auswertung wie in Abb. 4.3 (vorangegangenes Kapitel) gezeigt zu erstellen, benötigt man Daten in exakt dem folgenden Format:

```
;NN
BeMaTaC_L1_3.0;1056/8835
NoSta-D-TueBaDZ;1878/10830
```

Dies entspricht den nachfolgenden tabellarischen Daten.

Korpusname	Häufigkeit »NN«	Gesamtzahl STTS-Tags
BeMaTaC_L1_3.0	1056	8835
NoSta-D-TueBaDZ	1878	10830

Es handelt sich bei den in KoGra-R eingelesenen Daten also um ein Datenformat in CSV-Kodierung, bei welcher das Semikolon für eine Spaltengrenze steht. Das oben angegebene Beispiel entspricht also einer zwei-

spaltige Tabelle, in welcher in der Kopfzeile die ausgewertete Kategorie (NN) angegeben wird und in den darunterliegenden Zeilen das ausgewertete Korpus in der linken Spalte steht und in der rechten Spalte daneben zunächst die absoluten Häufigkeiten dieser Kategorie und, mit Backslash getrennt, der Normalisierungswert angegeben wird. Um die Standardauswertungen von KoGra-R eigenhändig zu erstellen, gehen Sie wie folgt vor.

Anleitung

- Laden Sie die unter der Internetadresse https://bit.ly/2CwsTOu verfügbaren Daten herunter.
- Laden Sie diese auf der Webseite http://kograno.ids-mannheim.de/ unter dem Menüpunkt »Freie Eingabe von Nutzer-definierten Tabellen« hoch (»Datei auswählen«).
- Betätigen Sie die Funktion »Tabelle prüfen«.
- Betätigen Sie auf der nächsten Seite die Funktion »Tabelle auswerten«.
- Wechseln Sie im geöffneten Zusatzfenster zur Option »eingegebene Daten«.
- Scrollen Sie hier durch verschiedenen Auswertungsoptionen. Die erzeugten Grafiken können herauskopiert werden. Berücksichtigen Sie erst einmal nur die Menüpunkte bis »Relative Werte (Diagramm, gruppiert)« und schauen Sie sich die Erläuterungen hierzu an.
- Um die Auswertung mit einem komplexeren Datensatz vorzunehmen, durchlaufen Sie diese Prozedur mit der unter https://bit.ly/2CvhbU4 beziehbaren Datei. Sie beinhaltet den Vergleich drei verschiedener Kategorien in zwei Korpora. Öffnen Sie die Datei, um zu sehen, wie die Daten hierfür angeordnet sein müssen.

4.5.3 | Ermittlung und Darstellung von Mittelwerten und Varianz

Die bisher behandelten Verfahren zur Darstellung von Häufigkeiten bestimmter Kategorien in Korpora werden häufig angewendet, bergen aber gewisse Probleme bzw. die Gefahr, falsche Schlussfolgerungen aus den Auswertungen zu ziehen. Normalisiert man einen absoluten Wert über das gesamte Korpus, so ist dies zwar generell ein Weg, Vergleichbarkeit mit einem anderen ausgewerteten Korpus zu schaffen, es bedeutet aber auch immer, dass man über die Gesamtheit der Korpusdaten gemittelt hat. Sofern das ausgewertete Phänomen bzw. die ausgewertete Kategorie aber in der Gesamtheit der Korpusdaten nicht relativ gleichmäßig verteilt ist, sind sämtliche Schlussfolgerungen problematisch, die suggerieren, dass der Normalisierungswert ein zuverlässiger Richtwert über die Gesamtheit der ausgewerteten Daten ist.

Untersuchungsszenario zur Erläuterung des Problems: In einem Korpus gesprochener Sprache soll ermittelt werden,

- wie viele gefüllte Pausen (Pausen innerhalb des Sprechflusses, gefüllt mit *ähm* oder ähnlichen Signalen) die Sprecher produzieren sowie
- wie häufig Schwa-Tilgungen bei Verben stattfinden.

Durch die Ermittlung solcher Auswertungsdaten können Rückschlüsse über das Wesen des entsprechenden gesprochenen Registers gezogen werden.

Das konkrete Ziel im gegebenen Szenario ist, das bereits erwähnte Korpus »BeMaTaC« (https://hu.berlin/bematac/), genauer das Subkorpus »BeMaTaC L1 Version 3.0« mit gesprochenen Dialogen deutscher Muttersprachler auf die beiden Kategorien hin auszuwerten. Das Korpus ist im ANNIS-Webinterface der Humboldt-Universität zu Berlin durchsuchbar (https://korpling.german.hu-berlin.de/annis3/), die Suche nach gefüllten Pausen (df=/f.*/) kann unter der Webadresse https://hu.berlin/bematac_anfrage_pausen, die Suchanfrage für Schwa-Tilgungen unter https://hu.berlin/bematac_anfrage_schwatilgung nachvollzogen werden. (Die letzte Anfrage ist komplexer, weil nach Verben gesucht wird, die ohne Schwa artikuliert und normalisiert mit Schwa-*e* geschrieben werden und entweder auf *-e* oder *-en* auslauten.)

Die Trefferzahl für gefüllte Pausen ist 272, die für die Tilgungen ist 84. In dem durchsuchten Korpus mit 12 Einzelgesprächen werden also 272 gefüllte Pausen produziert und bei 84 Verben das Schwa der letzten Silbe elidiert. Um hieraus einen aussagekräftigen Wert zu erzeugen, müssen die absoluten Werte an einem entsprechenden Normalisierungswert gemessen werden. Im ersten Fall der gefüllten Pausen kann dies durch die Gesamtzahl der Textwörter (Suchanfrage: dipl) erfolgen (potenziell kann man nach jedem Wort eine solche Pause anschließen). Im Fall der Schwa-Tilgungen muss der absolute Wert an der Zahl der standardisierten (normalisierten) Verbformen mit auslautendem *-e* bzw. *-en* gemessen werden (Suchanfrage: `pos=/V.*/ _=_ norm=/.*(en|e)/`). Der Normalisierungswert ist somit für die Pausen 9228 und für die Schwa-Tilgungen 258. Gemäß diesen Werten ergibt sich die Aussage, dass das Dialogkorpus pro 100 Bezugseinheiten gerundet 2,9 gefüllte Pausen und 32,6 Verben mit Schwa-Tilgungen aufweist. Dies ist in Abb. 4.7 visualisiert.

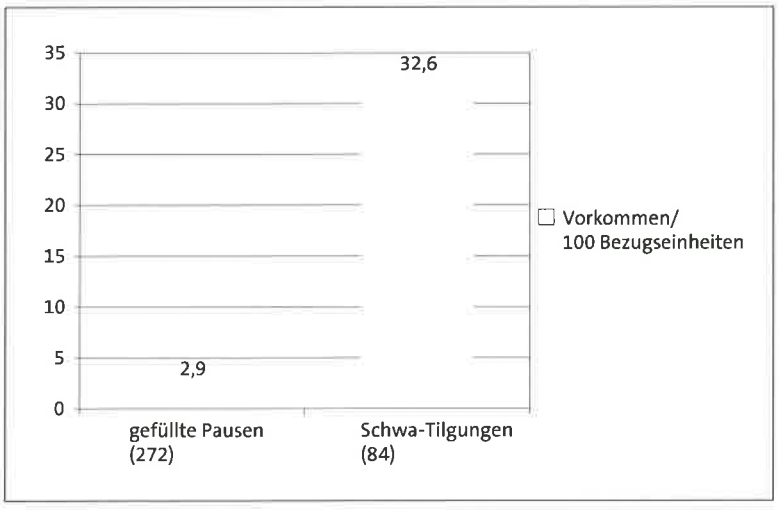


Abb. 4.7: Normalisierte Frequenzen für gefüllte Pausen und Schwa-Tilgungen bei Verben im Korpus »BeMaTaC L1 v3.0«

Nun stellt sich ~~nicht nur~~ die Frage, was man aus diesen Werten schlussfolgern darf. Denkbar wären Aussagen wie:

Gefüllte Pausen nehmen in spontansprachlichen Dialogen des Deutschen ca. 3 % des gesprochenen Texts ein. Elisionen von Schwa-Lauten bei Verben finden bei ziemlich genau einem Drittel der Verben mit elidierbarem Schwa statt.

Berücksichtigung der Varianz bei der Messung gemittelter Häufigkeiten: Bevor man sich fragt, ob man überhaupt aus dem Texttyp des Korpus (dem gesprochenen Register) auf einen größeren Bereich wie ›spontansprachliche‹ Dialoge‹ schließen darf, muss man prüfen, ob die ermittelten Werte überhaupt repräsentativ für das ausgewertete Korpus sind. Zu beachten ist, dass die angegebenen Normalisierungswerte Mittelwerte über das gesamte Korpus darstellen. Es kann sein, dass sie zustande kommen, weil sich die verschiedenen Sprecherinnen und Sprecher im Korpus so verhalten, wie diese Mittelwerte suggerieren, oder aber – in dem anderen Extrem – können sie das Resultat von stark abweichendem Sprecherverhalten sein und somit mit dem Verhalten der einzelnen Dialogpartner wenig zu tun haben. Um das zu überprüfen, kann man sich die Verteilung der gemessenen Kategorie über entsprechende Teile des Korpus hinweg anschauen und ermitteln, ob die gemessenen Mittelwerte eher zufällig erscheinen oder aus einem stabilen Trend heraus resultieren.

Betrachtet man das Verhalten der Sprecherinnen und Sprecher in den einzelnen Dialogen des Korpus, so ergibt sich für die Produktion der Pausen ein relativ heterogenes Bild (s. Abb. 4.8).

Die Daten für eine Übersicht wie in Abb. 4.8 lassen sich gewinnen, indem in ANNIS nach der Sucheingabe für gefüllte Pausen (s. o.) die Exportfunktion »WekaExoprter« genutzt wird und die jeweilige Zugehörigkeit des Treffers zu einem bestimmten Dokumentnamen (Eingabe: »metakeys = document« im »Parameters«-Eingabefeld) mit exportiert wird. Streudaten wie die in Abb. 4.8 abgebildeten lassen sich besser in einem Streudiagramm wie in Abb. 4.13 (s. u.) darstellen.

Der Normalisierungsfaktor beträgt in Abb. 4.8 nicht 100, sondern 1000, damit die Skalierungen bei der Auswertung der gefüllten Pausen und die

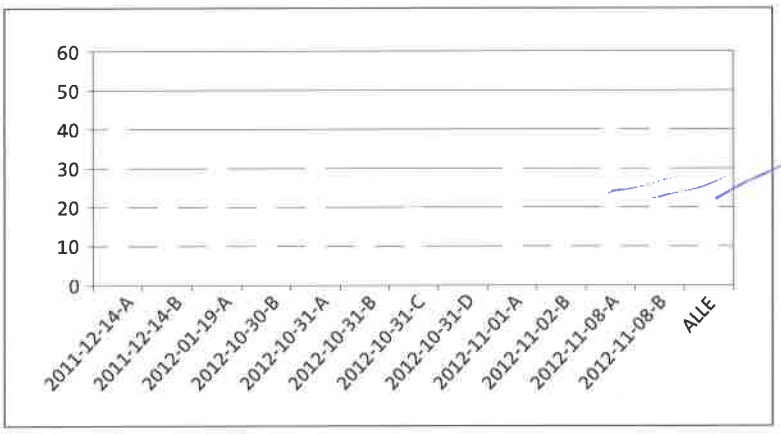


Abb. 4.8: Mittelwerte für die Verwendung gefüllter Pausen pro 1000 Textwörter in den einzelnen Dialogen des BeMaTaC-Korpus sowie der Mittelwert für die Gesamtdatenmenge

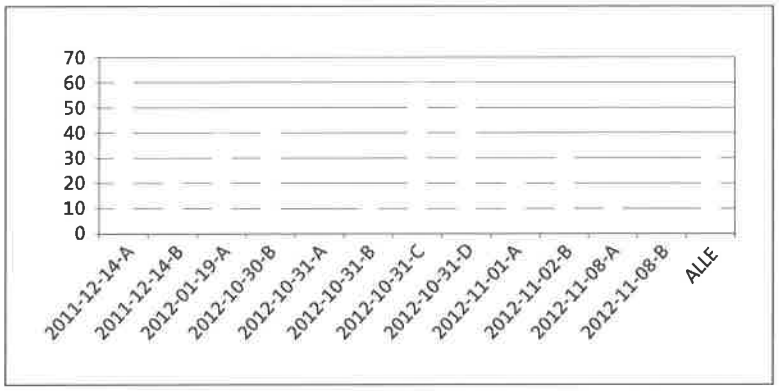


Abb. 4.9: Mittelwerte für die Häufigkeit von Schwa-Tilgungen bei Verben pro 100 Verben mit tilgbarem Schwa in den einzelnen Dialogen des BeMaTaC-Korpus sowie der Mittelwert für die Gesamtdatenmenge

der Schwa-Tilgungen vergleichbar sind. Dies ist nötig für den Vergleich der Varianz in den Daten. Abb. 4.8 zeigt, dass die relative Verwendungshäufigkeit in den einzelnen Dialogen des ausgewerteten Korpus mehr oder weniger stark von dem Gesamtmittelwert abweichen, wobei das Phänomen in allen Texten auftritt und auch nach oben hin keine extremen Ausreißer existieren. Der extremste Mittelwert liegt bei knapp unter fünf gefüllten Pausen pro 100 Textwörter. Die stärkste Abweichung von dem Gesamtmittelwert ist der Dialog »2012-10-31-D« mit nur etwa einem Drittel der Vorkommen. Der stärkste Unterschied zwischen den einzelnen Dialogen besteht zwischen dem Dialog »2012-10-31-D« mit normalisiert 9,3 Vorkommen und dem Dialog »2012-11-08-B« mit gemittelt 48,5 Vorkommen, womit die größte Varianz in der Produktion von gefüllten Pausen ungefähr den Faktor 5 beträgt.

Für die zweite Auswertung der Schwa-Tilgungen ergibt sich das in Abb. 4.9 dargestellte Bild.

In Abb. 4.9 ist die Heterogenität noch krasser als in Abb. 4.8, denn bezogen auf die Schwa-Tilgungen existiert das untersuchte Phänomen nicht einmal in allen Dialogen (obwohl in allen Dialogen Verben mit elidierbaren Schwa-Lauten auftreten). Hier ist fraglich, ob der über das Korpus errechnete Mittelwert überhaupt aussagekräftig für das Verhalten der einzelnen Sprecherinnen und Sprecher in den Gesprächen ist.

Es gibt verschiedene Verfahren, die aufgezeigte Datenstreuung in einem Korpus statistisch aufzuzeigen. Zunächst werden die grundlegenden Darstellungsmethoden an einem einfacheren, konstruierten Beispiel erläutert, anschließend werden die eingeführten Verfahren auf die Daten aus dem BeMaTaC-Korpus bezogen.

Die Berechnung der Mittelwertabweichung und der Standardabweichung sind Verfahren zur Bestimmung der Streuung um einen Mittelwert; es sind Maße für die Streuung der einzelnen Datenpunkte. Stellen Sie sich z. B. vor, Sie werten fünf Subkorpora auf die Verwendung von Schimpfwörtern aus, die in diesem Auswertungsbeispiel größenmäßig absolut vergleichbar sind. Das Ergebnis der Zählung sei das in Tab. 4.2 dargestellte.

Im Durchschnitt werden nach dem arithmetischen Mittel 22 Schimpf-

Tab. 4.2: Fiktive Verteilung von Schimpfwörtern in fünf gleichgroßen Korpora

Subkorpus 1	Subkorpus 2	Subkorpus 3	Subkorpus 4	Subkorpus 5
11	45	3	20	31

wörter verwendet ((11 + 45 + 3 + 20 + 31) ÷ 5). Man sieht aber, dass die Streuung zwischen den einzelnen Subkorpora extrem hoch ist. Die Mittelwertabweichung MA von 22 errechnet sich gemäß dem Beispiel wie folgt:

$$MA = \frac{|11 - 22| + |45 - 22| + |3 - 22| + |20 - 22| + |31 - 22|}{5}$$

Diese Pipes sollten weiter hinter Teilungslinie gehen.

Die vertikalen Striche stehen dafür, dass immer der Betrag der Differenz, also der Absolutwert bzw. positive Zahlenwert genommen wird (es ist egal, ob die Abweichung positiv oder negativ ist).

Das Ergebnis der Gleichung und somit die Mittelwertabweichung der einzelnen Werte beträgt 12,8. In Worten ausgedrückt, weichen also die Werte der einzelnen Subkorpusauswertungen durchschnittlich um den Absolutwert 12,8 von ihrem Mittelwert 22 ab. Dies entspricht mehr als der Hälfte des Mittelwerts selbst.

Die Standardabweichung (s) ist die Wurzel aus der Varianz. Sie errechnet sich gemäß den gegebenen Daten wie folgt:

$$s = \sqrt{\frac{(11 - 22)^2 + (45 - 22)^2 + (3 - 22)^2 + (20 - 22)^2 + (31 - 22)^2}{5}}$$

Die Varianz (der Bruch ohne die Wurzel) beträgt 219,2. Das Ergebnis für die Standardabweichung ergibt gerundet 14,8. Aufgrund der relativ hohen Abweichungen der einzelnen Datenpunkte liegt die Standardabweichung also um den Wert 2 höher als die Mittelwertabweichung. Für beide Werte gilt die Schlussfolgerung, dass die Streuung der Daten sehr hoch ist.

Die Berechnung von Konfidenzintervallen ist ein weiteres Verfahren für die Bestimmung der Datenstreuung um einen Mittelwert herum. Es wird anhand der gemessenen Datenpunkte und einem bestimmten Wahrscheinlichkeitswert. Das Standard-Konfidenzintervall mit einer 95 %-Wahrscheinlichkeit für die Schimpfwortverteilung in den fünf Subkorpora ist in Abb. 4.10 dargestellt.

verstellt

Zahlenmäßig ausgedrückt, kann man das Konfidenzintervall wie folgt abbilden.

obere Grenze	Mittelwert	untere Grenze
42,6	22,0	1,4

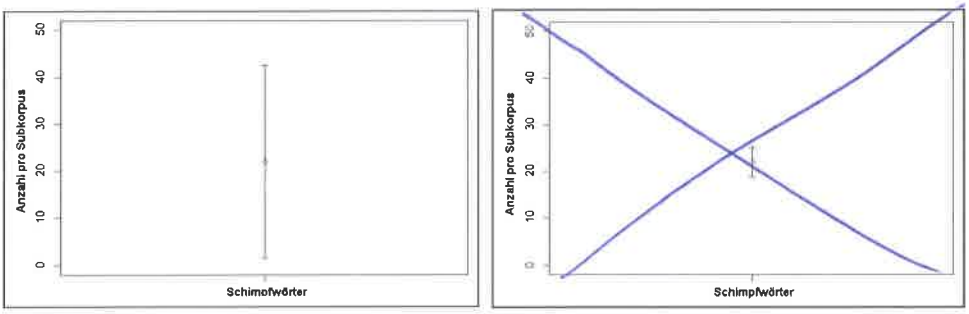


Abb. 4.10: R-Plot mit 95 %-Konfidenzintervall für die fiktive Verteilung von Schimpfwörtern (s. Tab. 4.2)

Grob gesagt, beantwortet das Konfidenzintervall die Frage, wie sehr ein berechneter Mittelwert die einzelnen Datenpunkte repräsentiert, denen er zugrunde liegt. Je größer der Bereich des Konfidenzintervalls (innerhalb derselben Datenpopulation) ist, desto größer ist die Streuung der Datenpunkte um den Mittelwert.

Schauen wir, was geschieht, wenn die Datenpunkte homogener verteilt sind und (in derselben Datenpopulation) näher am Mittelwert liegen: Nehmen wir an, in fünf Subkorpora sähe die Verteilung um den Mittelwert 22 wie in Tab. 4.3 aus.

Subkorpus 1	Subkorpus 2	Subkorpus 3	Subkorpus 4	Subkorpus 5
19	20	25	22	24

Tab. 4.3: Alternative Verteilung von Schimpfwörtern in fünf gleichgroßen Korpora

Unter diesen Bedingungen verändert sich das Bild der Varianz vollständig (s. Abb. 4.11).

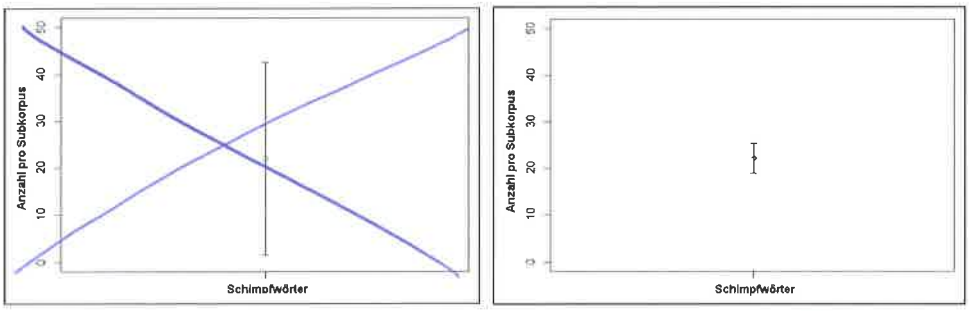


Abb. 4.11: R-Plot mit 95 %-Konfidenzintervall für die alternative Verteilung von Schimpfwörtern (s. Tab. 4.3)

Die Daten des Konfidenzintervalls in Abb. 4.11 sind in Tab. 4-4 angegeben.

obere Grenze	Mittelwert	untere Grenze
25,2	22,0	18,8

Tab. 4-4: Werte für das in Abb. 4.11 dargestellte Konfidenzintervall

Im Gegensatz zum vorherigen Szenario ergibt sich derselbe Mittelwert aus deutlich weniger gestreuten Daten, was zu einem deutlich kleineren Konfidenzintervall führt.

Wenden wir die Konzepte zur Bestimmung der Datenstreuung auf das authentische Auswertungsszenario der Analyse von gefüllten Pausen und Schwa-Tilgungen im BeMaTaC-Korpus an, so ergeben sich die Werte und Visualisierungen in Tab. 4.5 und Abb. 4.12.

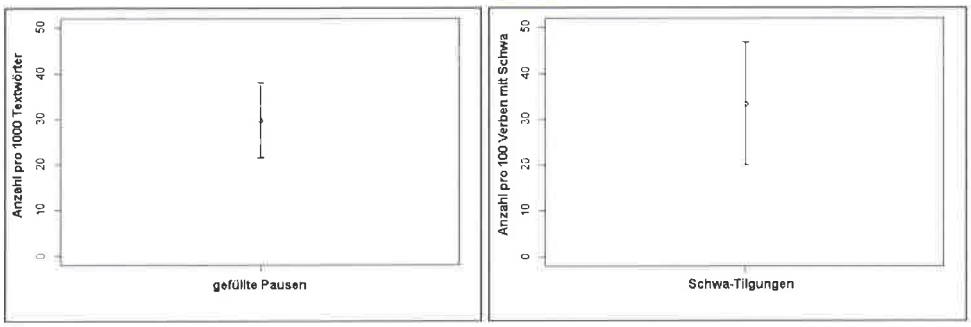


Abb. 4.12: Gegenüberstellung der 95 %-Konfidenzintervalle zu den Mittelwerten der gefüllten Pausen und der Schwa-Tilgungen im BeMaTaC-Korpus

	gefüllte Pausen	Schwa-Tilgungen
Mittelwertabweichung	10,6	16,9
Standardabweichung	12,4	21,0

Tab. 4.5: Werte für die Mittelwert- und Standardabweichung der oben eingeführten Variablen »gefüllte Pausen« und »Schwa-Tilgungen« im Korpus BeMaTaC (s. Abb. 4.8 und Abb. 4.9)

Die vorangegangenen Auswertungen zeigen, dass alle Verfahren zur Berechnung und Darstellung der Varianz für das Auftreten der Schwa-Tilgungen eine deutlich höhere Varianz ausgeben als für die gefüllten Pausen, so dass man als Schlussfolgerung festhalten muss, dass die Produktion gefüllter Pausen im Korpus ein stabileres Phänomen ist als das Phänomen von Schwa-Tilgungen. Während der normalisierte Mittelwert für gefüllte Pausen noch mit einiger Sicherheit um 30 Vorkommen pro 1000 Textwörter angegeben werden kann, erscheint die Angabe des errechneten Mittels von 32,6 Schwa-Tilgungen pro 100 Verben mit elidierbarem Schwa die einzelnen Datenpunkte wegen ihrer stärkeren Streuung deutlich schlechter zu beschreiben.

Diese Unterschiede in der Varianz sind keinesfalls als negativer Faktor bei der Auswertung zu erachten, sondern sind interessante und wesentliche sprachliche Eigenschaften und sollten bei der Korpusanalyse unbedingt berücksichtigt werden.

Typen von Mittelwerten – arithmetisches Mittel und Median: Eine alternative Darstellung der gefüllten Pausen im BeMaTaC-Korpus ist die in Abb. 4.13 dargestellte (sie basiert auf denselben Daten wie die bisherigen Statistiken).

Die Grafik in Abb. 4.13 wurde mit dem Statistikprogramm RStudio (http://www.rstudio.com/products/rstudio/) und dem frei verfügbaren R-Paket »ggplot2« (https://ggplot2.tidyverse.org/) erstellt und stellt wie Abb. 4.8 die Streuung der durchschnittlichen Häufigkeiten von gefüllten Pausen in den einzelnen Dialogen des BeMaTaC-Korpus dar. Was hier fehlt, ist der Mittelwert, der sich aus diesen Datenpunkten ergibt. Eine Möglichkeit der Bestimmung eines Mittelwerts ist die Berechnung des ›arithmetischen Mittels‹. Dieses errechnet sich durch die addierten Werte

aller Datenpunkte, geteilt durch die Anzahl der Datenpunkte. Nach dieser Definition ist der Mittelwert von 10, 20 und 24 gleich 10 + 20 + 24, geteilt durch drei, also 18. Bezogen auf die gefüllten Pausen in BeMaTaC beträgt das arithmetische Mittel den Wert 29,4.

Alternativ kann man den Mittelwert von gestreuten Datenpunkten angeben, indem man den mittleren Datenpunkt errechnet. Gemäß dem Beispiel von 10, 20 und 24 liegt dieser bei 20. Bei einer geraden Anzahl von Datenpunkten wie im Fall der gefüllten Pausen im BeMaTaC-Korpus (es handelt sich um 12 Dialoge, also 12 Datenpunkte) wird der Median durch die Mitte zwischen den beiden innen liegenden Punkten (die Mitte zwischen dem sechsten und siebten Punkt im BeMaTaC-Szenario) angegeben. Abb. 4.14 zeigt die Lage des arithmetischen Mittelpunkts (rot; die Markierung liegt nur zufällig über einem Datenpunkt) und der Lage des Medians (blau, die Markierung liegt exakt zwischen dem sechsten und dem siebten Datenpunkt).

Der wesentliche Effekt des Medians gegenüber dem arithmetischen Mittel ist, dass extreme Ausreißer nicht so sehr ins Gewicht fallen. Man kann nicht pauschal sagen, welche Mittelwertangabe die bessere ist. Während das arithmetische Mittel angibt, wo in einem Spektrum von Datenpunkten der ›Durchschnittwert‹ liegt, besagt der Median, an welcher Stelle der zentrale Datenpunkt liegt.

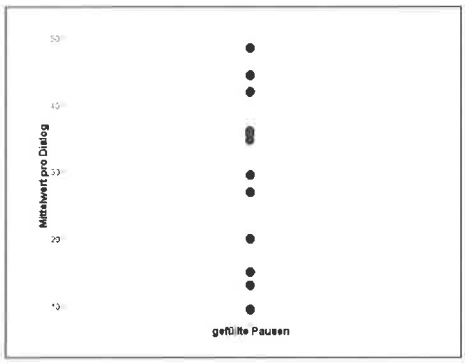


Abb. 4.13: Abbildung der gefüllten Pausen im BeMaTaC-Korpus, aufgeteilt auf einzelne Dialoge

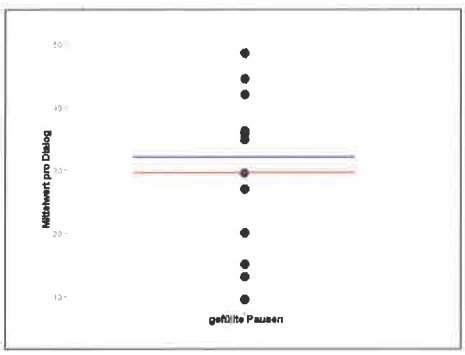


Abb. 4.14: Abbildung der gefüllten Pausen im BeMaTaC-Korpus, aufgeteilt auf einzelne Dialoge, mit Durchschnittswert (rot) und Median (blau)

4.5.4 | Berechnung statistischer Signifikanz

Definition

> Die **statistische Signifikanz** ist ein Konzept zur Bestimmung der Aussagekraft von Messungen wie z. B. Korpusauswertungen. Sie ergibt sich aus einem Wahrscheinlichkeitswert für die Zufälligkeit der vorgenommenen Messungen: Ist der Wert hoch, so wird die Aussagekraft der Messungen für gering befunden, ist der Wert gering, so wird die Aussagekraft für hoch befunden.

Im vorigen Abschnitt wurde gezeigt, dass die Aussagekraft eines Werts wie z. B. dem Mittelwert u. a. von der Größe der Varianz der Datenpunkte abhängt, auf die sich der Mittelwert bezieht. Ist das Spektrum der Datenpunkte sehr groß, so scheint der Mittelwert die einzelnen Datenpunkte weniger gut zu beschreiben und hat damit relativ wenig Aussagekraft für die Daten; ist das Spektrum der Datenpunkte gering, drückt der Mittelwert einen allgemeinen Trend aus und besitzt eine hohe Aussagekraft. Wir erwarten für Messungen, denen in diesem Sinn eine geringere Aussagekraft beizumessen ist, einen weniger signifikanten Wert, und für Messungen mit höherer Aussagekraft einen höheren Signifikanzwert. Signifikanz gibt also ein statistisch berechnetes Urteil über die Aussagekraft von Messungen ab. Dies geschieht über eine Wahrscheinlichkeitsrechnung, aus der ein Wert (der sog. ›p-Wert‹, wobei ›p‹ für engl. *probability*, Wahrscheinlichkeit, steht) hervorgeht. Wenn der Wert niedrig ist, wird der Messung eine hohe Aussagekraft beigemessen; wenn er hoch ist, wird der evaluierten Messung eine geringe Aussagekraft zugeschrieben, sie gilt als weniger (oder nicht) verallgemeinerbar.

Der Wert p strebt auf der einen Seite der Werteskala gegen null und die bewertete Messung ist bei null maximal signifikant. Auf der anderen Seite der Skala strebt p gegen 1 und ist im Fall von 1 minimal signifikant.

Da das Spektrum, das p annehmen kann, kontinuierlich ist, muss ein Schwellenwert festgelegt werden, ab dem das Testergebnis so interpretiert wird, dass die überprüfte Messung zuverlässig bzw. nicht zuverlässig ist. Der gewöhnliche Schwellenwert liegt bei $p = 0{,}05$, d. h. alle Werte, die kleiner als dieser sind, werden als statistisch signifikant interpretiert; alle Werte, die darüber liegen, werden als nicht signifikant interpretiert, was dazu führt, dass die überprüften Messergebnisse als nicht aussagekräftig aufgefasst werden.

Untersuchungsszenario und Testbeispiele: Es ist nicht sinnvoll, gewisse Testverfahren für Signifikanzniveaus zu nennen und vorzugeben, es handele sich um Standardverfahren. Eine solche Vorgehensweise ist deshalb nicht zielführend, weil sich der statistische Test nach dem Ziel der angestrebten Studie richten muss. Im Grunde benötigt jede Studie bzw. jede Fragestellung eine auf sie angepasste Zusammenstellung an methodischen Ansätzen, einschließlich der statistischen Verfahren. In dem folgenden Untersuchungsszenario soll ein häufig auftretendes grundlegendes Anliegen durchgespielt werden: Man fragt sich, ob sich

die Verwendung einer bestimmten sprachlichen Kategorie in zwei Varietäten, vertreten durch zwei Korpora A und B, nachweislich unterscheidet.

Das folgende Beispiel ist eine Erweiterung der Auswertung des BeMaTaC-Korpus im vorangegangenen Kapitel 4.5.3 zur Ermittlung und Darstellung von Mittelwerten und der Varianz von Kategorien in Korpora. Dort wurde die Produktion von gefüllten Pausen in den Dialogen, die im Korpus annotiert wurden, auf deren mittlere Verteilung und die Varianz ihres Auftretens über die verschiedenen Dialoge hinweg untersucht. Das untersuchte Korpus (BeMaTaC L1) besteht aus Dialogen deutscher Muttersprachlerinnen und Muttersprachler. Es existiert ein zweites Korpus mit Dialogen fortgeschrittener Lernender des Deutschen als Fremdsprache (BeMaTaC L2). Die Kategorie der gefüllten Pausen kann als eine Kategorie angesehen werden, die relevant für das Deutsche als Lernersprache ist, denn sie kann als Hinweis auf das Merkmal Flüssigkeit angesehen werden, ~~Der~~ als einer der wesentlichen Faktoren im Spracherwerb gilt. Eine interessante Hypothese, die anhand der BeMaTaC-Korpusdaten aufgestellt und überprüft werden kann, ist:

> Trotz ihrer Fortgeschrittenheit produzieren die Lernenden des Deutschen als Fremdsprache im BeMaTaC-L2-Korpus signifikant mehr gefüllte Pausen ~~auf~~ als die Muttersprachlerinnen und Muttersprachler des BeMaTaC-L1-Korpus. Dies weist auf eine entsprechend verminderte Sprachflüssigkeit der Lernenden des Deutschen als Fremdsprache gegenüber den deutschen Muttersprachlerinnen und Muttersprachler hin.

Der erste Schritt zur Überprüfung dieser Annahme ist der Vergleich normalisierter Mittelwerte der zwei Korpora. Der Mittelwert für gefüllte Pausen im BeMaTaC-L1-Korpus wurde bereits in Kapitel 4.5.3 ermittelt: Er basierte auf 272 absoluten Treffern (Suchanfrage »df = /f.*/« auf dem Korpus »BeMaTaC_L1_3.0« im ANNIS-Suchinterface: https://hu.berlin/bematac_anfrage_pausen) und einer Normalisierungsgröße von 9228 artikulierten Worteinheiten (Suchanfrage: »dipl«) im Korpus. Somit betrug die normalisierte Häufigkeit von gefüllten Pausen im BeMaTaC-L1-Korpus 2,9 Pausen pro 100 Textwörter. Um eine unmittelbar vergleichbare Größe für die Häufigkeit gefüllter Pausen im Lernerkorpus »BeMaTaC_L2_3.0« zu erlangen, wählt man dieses Korpus an (und ggf. das L1-Korpus ab) und führt dieselben Suchanfragen aus. Die Suche nach gefüllten Pausen erzielt 425 Suchtreffer, die Normalisierungsgröße beträgt 9810 (dies ist die Anzahl an Textwörtern, ermittelt durch die Suchanfrage »dipl«). Somit ist der vergleichbare (normalisierte) Wert für das L2-Korpus gerundet 4,3 Pausen pro 100 Textwörter, also ca. 50 % häufigere Pausen als im L1-Korpus. Visualisiert ergibt sich das Diagramm in Abb. 4.15.

Diese Verhältnisse spiegeln die Realität innerhalb der ausgewerteten Korpusdaten wider. Eine Berechnung der statistischen Signifikanz über den ausgewerteten Daten sagt uns, ob über die Korpusdaten selbst verallgemeinern dürfen. Konkret sollen folgende Fragen beantwortet werden:

- Haben die ermittelten Unterschiede etwas mit der Welt außerhalb der Daten, in der sie gemessen wurden, zu tun?

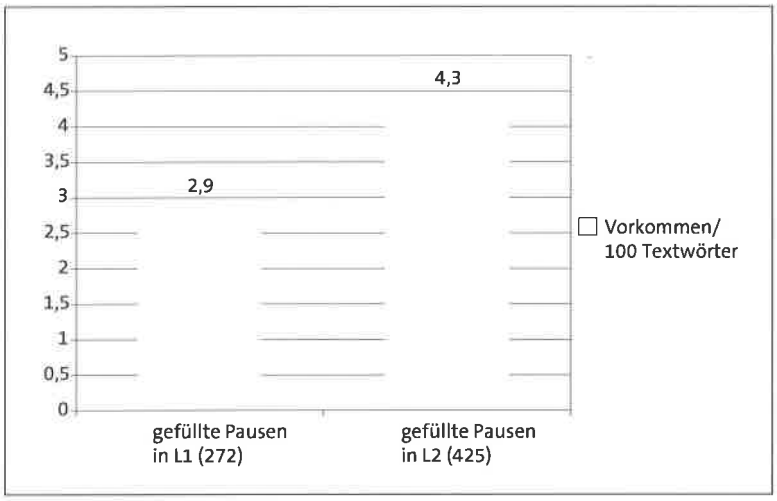


Abb. 4.15: Auswertung von Pausen im BeMaTaC-Korpus, L2 vs. L1 (normalisierte Werte)

- Haben die gemessenen Verhältnisse Bestand, wenn ein weiteres Mal muttersprachliche und nichtmuttersprachliche Probandinnen und Probanden untersucht würden?

Wie bereits erläutert, werden diese Fragen beantwortet, indem ein Wert für die Zufallswahrscheinlichkeit p ermittelt wird.

Gleichheitstests: Der einfachste Weg, dies zu tun, ist, die Größe der einzelnen Stichproben in Form der Normalisierungswerte (im konkreten gegebenen Fall 9228 Textwörter im L1-Korpus und 9810 im L2-Korpus) mit der Anzahl der jeweils gezählten Fälle (im gegebenen Fall 272 gefüllte Pausen im L1-Korpus und 425 im L2-Korpus) in einer mathematischen Gleichung in Beziehung zu setzen, so dass sich aus den Relationen der jeweiligen Anzahlen des beobachteten Phänomens und der jeweiligen Stichprobengröße ein Wahrscheinlichkeitswert dafür ergibt, dass die gemessene Relation zufällig vorliegt. Dies machen verschiedene Testverfahren, die man unter dem Namen ›Gleichheitstests für Stichprobenverhältnisse‹ zusammenfassen könnte.

In dem dargestellten Sinn anwendbar sind der Z-Test (Zweistichprobentest für unabhängige Stichproben) oder der Chi-Quadrat-Test (Homogenitätstest). In dem Statistikprogramm R (http://www.r-project.org/) bzw. der Nutzeroberfläche für R, RStudio (http://www.rstudio.com), steht der »prop.test« zur Verfügung (https://bit.ly/2FiMBxR), mit dem man unter Eingabe der besagten Größen einen Signifikanzwert ermitteln kann. Der Test gibt anhand absoluter Frequenzen zu einem Phänomen in zwei Datenmengen (Populationen) und der dazugehörigen Normalisierungsgrößen einen Chi-Quadrat-Wert aus, der in einen p-Wert umgerechnet wird. Um den Test durchzuführen, benötigen wir gemäß unserem Anwendungsszenario nur die absoluten Werte 272 und 425 für die Gesamtzahl der gefüllten Pausen in den BeMaTaC-Korpora L1 und L2 sowie die Normalisierungsgrößen 9810 und 9228, die der Gesamtzahl der Textwörter in den Korpora entsprechen. Es existieren verschiedene online ver-

fügbare Nutzeroberflächen (Eingabemasken), die Signifikanzwerte anhand von eingegebenen Werten ermitteln.

Die in der RStudio-Konsole erforderlichen Daten für den ›prop.test‹ gemäß dem gegebenen Auswertungsszenario sind:

```
prop.test(c(272,425),c(9810,9228))
```

Es werden also zunächst in der ersten Klammer die absoluten Anzahlen des gemessenen Phänomens und in der zweiten Klammer die Normalisierungsgrößen eingetragen. Als Ergebnis liefert das Programm für p einen hochsignifikanten Wert (4.5e-07 bzw. $4{,}5 \times 10^{-7}$ oder 0,00000045 und somit ein Wert weit unter 0,05 Zufallswahrscheinlichkeit), der besagt, dass die gemessenen Fälle der gefüllten Pausen in den beiden Stichproben mit an Sicherheit grenzender Wahrscheinlichkeit nicht aus derselben Grundgesamtheit stammen.

Einige Forscherinnen oder Forscher würden dies so interpretieren, dass die Ergebnisse der Auswertung über die Korpusdaten hinaus verallgemeinerbar sind. Das Problem bei statistischen Verfahren wie den angesprochenen, die lediglich Mittelwerte und Bezugsgrößen verrechnen, ist, dass sie die Verteilung der Daten nicht miteinbeziehen können und den Berechnungen meistens Annahmen über die Verteilung zugrunde liegen, die für die tatsächlichen Daten nicht gelten. In Kapitel 4.5.3 haben wir jedoch gesehen, dass die gemessenen Phänomene mehr oder weniger stark variieren. Die Schlussfolgerung war, dass Mittelwerte, die auf starker Varianz beruhen, weniger zuverlässig sind als Mittelwerte, die sich aus näher zusammenliegenden Datenpunkten ergeben. In einer statistischen Berechnung, in der wir lediglich Mittelwerte und Werte für die Größe der Bezugsmenge einspeisen, wird dieser Gesichtspunkt nivelliert. Deshalb muss die Varianz als ein wesentlicher Faktor in die Kalkulation miteinbezogen werden.

In dem gegebenen Auswertungsbeispiel lassen sich die gemessenen Unterschiede mitsamt der Varianz in den Daten über die einzelnen Dialoge hinweg wie in Abb. 4.16 und Abb. 4.17 visualisieren.

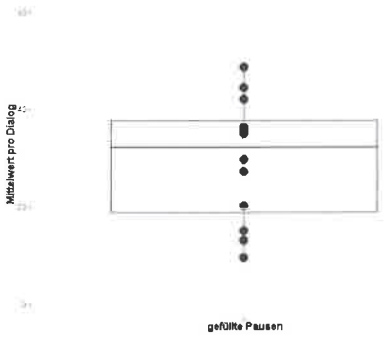


Abb. 4.16: Streuung der Mittelwerte für gefüllte Pausen pro Dialog im BeMaTaC-Korpus L1_3.0

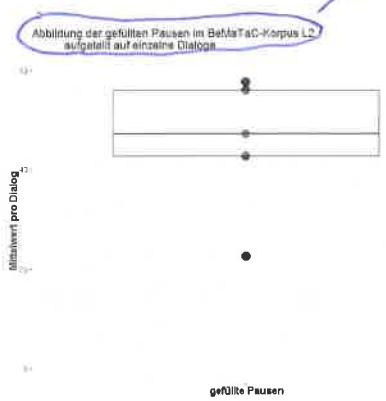


Abb. 4.17: Streuung der Mittelwerte für gefüllte Pausen pro Dialog in BeMaTaC-Korpus L2_3.0

Abb. 4.16 und Abb. 4.17 zeigen sowohl den jeweiligen Mittelwert (Median) für das gemessene Phänomen (die jeweils mittlere schwarze Linie innerhalb der Boxen) als auch die Streuung der Daten durch die einzelnen Datenpunkte, die die gemittelte Häufigkeit des Phänomens in den einzelnen Dialogen der Korpora abbilden. Mithilfe der Größe der Boxen werden außerdem anhand der Streuung der Datenpunkte 95 %-Konfidenzen nach oben und unten vom Mittelwert aus angegeben. Überlappen die Intervalle nicht, ist dies für die Zuverlässigkeit der Mittelwertunterschiede ein gutes Zeichen.

Durch die in den beiden Grafiken Abb. 4.16 und Abb. 4.17 gezeigten Datenpunkte lassen sich Signifikanztests durchführen, in deren Berechnung die Varianz innerhalb der Korpusdaten miteinbezogen wird. Solche Signifikanzberechnungen sind somit deutlich aussagekräftiger als die oben genannten Berechnungen, die bestimmte Vorannahmen zur Datenverteilung machen. Der Mann-Whitney-U-Test (auch Wilcoxon-Mann-Whitney-Test genannt) ist ein statistischer Test, der keine Vorannahmen zur Datenverteilung macht (ein sog. parameter- oder verteilungsfreier Test). Hier müssen die einzelnen Datenpunkte in die Berechnung eingespeist werden, so dass die Verteilung der Datenpunkte in die Berechnung einfließen kann. Dasselbe gilt für den t-Test (in unserem Auswertungsfall der Zweistichproben-t-Test), der einen gewissen Standard bei einem Auswertungsszenario wie dem hier vorgestellten darstellt.

Die Datenpunkte, die in die Berechnung der Tests einfließen, können in unterschiedlichem Format vorliegen (s. u. die Anleitung für Details zu den Datenformaten und der Durchführung der Tests in RStudio).

Als Testergebnis für den Mann-Whitney-U-Test erhält man den gerundeten p-Wert 0,048, der sehr knapp unter dem Signifikanzniveau von 0,05 liegt bzw. diesem Wert entspricht, wenn man auf zwei Nachkommastellen rundet. Hinsichtlich des oben genannten Ergebnisses des ›prop.test‹ ist aufgrund der berücksichtigten Abhängigkeit der Daten der Signifikanzwert deutlich gesunken und liegt an der Grenze des Signifikanzniveaus.

Der Zweistichproben-t-Test liefert für dieselben Daten einen p-Wert von gerundet 0,071, der also außerhalb des Signifikanzniveaus liegt.

Die Tatsache, dass die drei erwähnten Testverfahren relativ unterschiedliche Signifikanzwerte ausgeben, verdeutlicht, wie problematisch die Ermittlung von Signifikanzen sein kann. Es sollte deutlich geworden sein, dass der verwendete Test bei der Nennung eines p-Werts stets mit angegeben werden muss, weil er für sich selbst nicht besonders aussagekräftig ist.

Um die erwähnten Statistiktests selber durchzuführen, halten Sie sich an die Anweisungen in der Anleitung.

Anleitung

Um in RStudio den erwähnten ›prop.test‹ durchzuführen, gehen Sie folgendermaßen vor:

- Laden Sie sich das Statistikprogramm RStudio herunter (https://www.rstudio.com/products/rstudio/download/), installieren Sie es und öffnen Sie das Programm.

- Ermitteln Sie für den Vergleich der Häufigkeiten eines Phänomens (einer Kategorie) in zwei Datenmengen bzw. Korpora die Gesamtmenge der jeweiligen Vorkommen (z. B. 272 und 425 im Anwendungsbeispiel der gefüllten Pausen) und die jeweiligen Bezugs- bzw. Normalisierungsgrößen (z. B. 9810 und 9228 im Anwendungsbeispiel).
- Tragen Sie diese vier Zahlen in das folgende Schema ein:
  prop.test(c(GESAMTZAHL1,GESAMTZAHL2),c(NORMALISIERUNGSGRÖßE1, NORMALISIERUNGSGRÖßE2))
- Geben Sie diese Formel in das Konsolenfenster von RStudio ein (links unten) und drücken Sie Enter.
- Der ausgegebene p-Wert soll etwas über die Verallgemeinerbarkeit der vorgenommenen Messung aussagen. Beachten Sie dabei allerdings bitte, dass die Annahmen des gewählten Tests über die Beschaffenheit der Daten und auch die Wahrscheinlichkeit für das Auftreten des gemessenen Phänomens nicht zu Daten passen, die aus Korpora stammen.

Um den Mann-Whitney-U-Test und den Zweistichproben-t-Test wie oben vorgestellt durchzuführen, gehen Sie folgendermaßen vor:

- Öffnen Sie RStudio.
- Wählen Sie oben im Menü die Option »File« > »New File« > »R Script«.
- Die Daten, die in die Signifikanzauswertung eingehen, sind die Werte für die durchschnittliche Verwendung von Pausen in den BeMaTaC-L1- sowie den BeMaTaC-L2-Dialogen. Diese sind:
  L1: `41,9; 26,9; 36,1; 29,4; 35,6; 12,9; 19,9; 9,3; 34,7; 14,9; 44,4; 48,5`
  (Es gibt 12 L1-Dialoge im Korpus)
  L2: `22,4; 47,1; 55,9; 42,7; 57,6` (Es gibt fünf L2-Dialoge im Korpus.)
  Sie können die Daten wie folgt in RStudio eingeben:
  ```
  L1 <- c(41.9,26.9,36.1,29.4,35.6,12.9,19.9,9.3,34.7,14.9,44.4,48.5)
  L2 <- c(22.4,47.1,55.9,42.7,57.6)
  ```
- Geben Sie darunter für die Durchführung des Mann-Whitney-U-Tests ein:
  ```
  wilcox.test(L1,L2)
  ```
- Geben Sie darunter für die Durchführung des Zweistichproben-t-Tests ein:
  ```
  t.test(L1,L2)
  ```
- Sie können alle eingegebenen Daten markieren und mit STRG-Enter verarbeiten lassen. Alternativ setzen Sie den Cursor in die oberste Befehlszeile und schicken Sie Zeile für Zeile mit STRG-Enter ab.
- Die relevanten Werte für die Auswertung sind die jeweiligen p-Werte in der ›Console‹ (unteres Fenster), für den Sie den Wert 0,05 als Grenze für das Signifikanzniveau ansetzen können.
- Das Script für RStudio mit den oben genannten Befehlen können Sie hier beziehen: https://bit.ly/2FrAjUV.

4.6 | Methoden für die Analyse einer bestimmten Varietät

Alternativ zum Vergleich von Frequenzen eines Phänomens oder mehrerer Phänomene in verschiedenen Datenpopulationen kann man ein Phänomen innerhalb ein und derselben Datenmenge untersuchen. Dies ist bei Fragestellungen sinnvoll, die auf die Zusammengehörigkeit, die Anziehung oder Abstoßung sprachlicher Elemente oder die Variation von Elementen in wechselnden sprachlichen Kontexten abzielen.

Diese Fragestellungen haben gemeinsam, dass es hier nicht um den Vergleich eines Phänomens in verschiedenen Varietäten, repräsentiert durch Korpora oder Subkorpora, handelt, sondern um Variation oder Abhängigkeiten von Elementen innerhalb derselben Datenmenge. Die anschließenden Kapitel geben einen Überblick über Konzepte, die zu dieser Grundausrichtung gehören.

4.6.1 | Variantenprofile für ein Korpus erstellen

Definition

> **Varianten** sind die zu einer bestimmten Variable gehörenden Ausprägungen. Eine **Variable** kann ein beliebiges linguistisches Phänomen sein, das verschiedene Erscheinungsformen (Varianten) zulässt.

Vergleichen Sie die Beispiele für Variablen und dazugehörigen Varianten in Tab. 4.6.

Tab. 4.6: Beispiele für Variablen und Varianten

Variable	Varianten
Verbstellung	Verb-erst, Verb-zweit, Verb-letzt
Personalpronomen	*ich, du, er/sie/es, wir, ihr, sie*
Definitheit	definit, indefinit
Temporalität	Vergangenheitsbezug, Gegenwartsbezug, Zukunftsbezug, zeitloser Bezug
Verb	Vollverb, Hilfsverb, Modalverb, Kopulaverb, Funktionsverb

Sofern die Varianten zu einer Variable bekannt sind und im Korpus ermittelt werden können, ergibt sich durch die Gegenüberstellung der Verteilungen der Varianten ein individuelles Profil zunächst für das untersuchte Korpus. Dies kann sehr aufschlussreich für die weitere linguistische Interpretation sein.

So lassen sich z. B. die wesentlichen Objekttypen in einem Korpus, in dem sie annotiert sind, wie dem TüBa-D/Z-Korpus (Version 8) gegenüberstellen (s. Abb. 4.18).

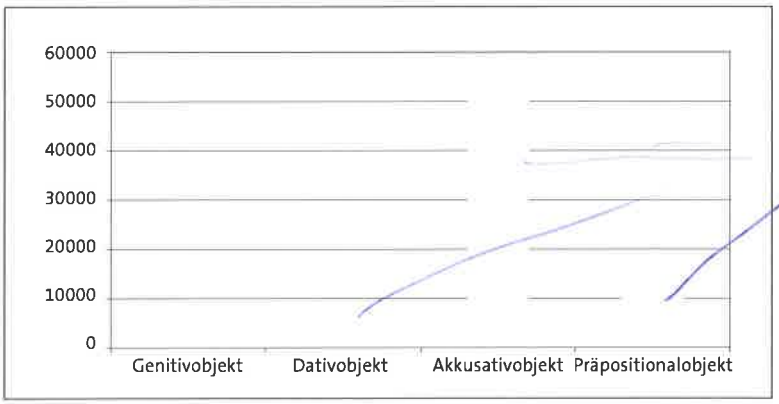


Abb. 4.18: Verteilung von Objekttypen im TüBa-D/Z-Korpus

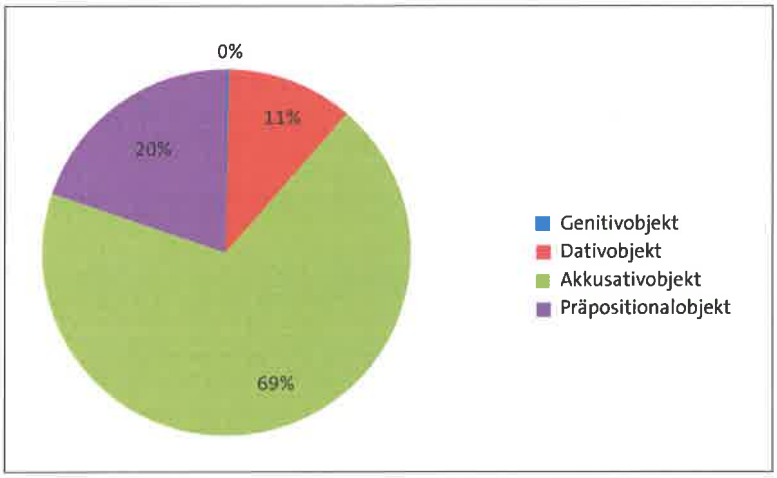


Abb. 4.19: Verteilung verschiedener Objekttypen im TüBa-D/Z-Korpus

Die in Abb. 4.18 dargestellten Daten lassen sich auch in einem Kreisdiagramm analysieren (anstelle absoluter Werte werden hier relative Prozentwerte angegeben; s. Abb. 4.19).

Für die historische Linguistik des Deutschen könnte eine Gegenüberstellung dieses Verteilungsprofils mit solchen von vergangenen Sprachstufen interessant sein, um den syntaktischen Wandel aufzuzeigen.

Im Folgenden werden weitere Szenarios für die Darstellung solcher Variantenprofile vorgestellt und die einzelnen Schritte zur Erstellung der Profile dargestellt.

Anwendungsszenarios: Exemplarisch sollen drei Variablen im Hinblick auf ihre Variantenverteilung untersucht werden: Personalpronomina und ihre einzelnen Lemmata, Präpositionen und ihre Kasus sowie Schreibungsvarianten im historischen Deutsch.

Formen von Personalpronomina in Korpora können Aufschluss über die Subjektivität der Sprache und die Adressatenbezüge im Text geben. In persönlichen Texten wie informellen E-Mails erwartet man mehr Anrede-

pronomina und Selbstbezüge durch *ich*, während in Sachtexten mehr Pronomina der dritten Person zu erwarten sind. Interessant ist die Verteilung in bestimmten Varietäten, in denen entscheidende Parameter wie die Narrativität oder der Adressatenbezug variieren können, wie z. B. in Romantexten mit hohem Anteil wörtlicher Rede und Protagonisteninteraktion sowie mündlichen politischen Debatten.

Um die Vorkommen der einzelnen Lemmata von Personalpronomen pro Korpus vergleichend nebeneinanderzustellen, muss man die Variable »Personalpronomen« suchen und die Treffer nach ihrem Lemma geordnet auflisten. Ein Normalisieren der Daten ist dann nötig, wenn die Aufstellungen in zwei Korpora miteinander verglichen werden sollen.

In allen in Kapitel 3.1.2 vorgestellten Suchsystemen sowie im COSMAS-II-Interface des IDS Mannheim (s. Kap. 3.2.2) ist eine Aufstellung von bestimmten Varianten zu einer Suchvariable unkompliziert machbar. Im Folgenden wird ein exemplarischer Lösungsweg für das ANNIS-Suchinterface (https://hu.berlin/annis-intro) dargestellt. Folgen Sie den einzelnen Arbeitsschritten der Anleitung, um den Lösungsweg selber nachvollziehen zu können.

Anleitung

- Wählen Sie das Korpus »Parlamentsreden_Deutscher_Bundestag« aus.
- Formulieren Sie die folgende Suchanfrage:
  `POS="PPER" _=_ LEMMA`
- Sie erhalten so 126.828, Treffer. Wählen Sie die Funktion »More« > »Frequency Analysis«. Löschen Sie den Wert »1 – POS« aus der Übersicht der Variablen, auf die zugegriffen werden kann (markieren Sie hierfür die Zeile mit der Variable und wählen Sie »Delete selected row(s)«). Wählen Sie anschließend »Perform frequency analysis«.
- Kopieren Sie die tabellarische Übersicht durch die Funktion »Download as CSV« und speichern oder öffnen Sie die exportierte Datei.
- Kopieren Sie das Ergebnis nach LibreOffice (oder OpenOffice) Calc oder Microsoft Excel.
- Sortieren Sie die Zeilen in der Reihenfolge der Werte »ich« – »du« – »er« – »sie« – »es« – »wir« – »ihr« – »Sie|sie«. (Behandeln Sie den achtmal vergebenen Annotationswert »er|er|es« als Annotationsfehler oder ordnen Sie mittels der Suche nach genau diesem Wert die in der Kategorie enthaltenen Treffer den korrekten Werten zu.)
- Markieren Sie in den zwei entsprechenden Spalten die Werte und Zahlen und wählen Sie anschließend »Einfügen« > »Diagramm...« > »Säulen« bzw. (in Excel) »Einfügen« > »(Diagramme) Säule«. Erzeugen Sie zusätzlich ein Kreisdiagramm, indem Sie die Daten erneut markieren und den entsprechenden Diagrammtyp auswählen.
- Erstaunlicherweise ist das relativ am häufigsten gebrauchte Lemma *wir*. (Die Gründe hierfür können durch eine qualitative Sichtung der entsprechenden Belege ermittelt werden.)
- Sie können diese relative Aufstellung nun mit Aufstellungen zu anderen Korpora vergleichen. (Beachten Sie hierbei, dass in den meisten anderen Korpora die Suchvariable für Wortarten »pos« und die für Lemmata »lemma« heißt.)

Wenn die jeweiligen Lemmawerte zwischen zwei Korpora direkt miteinander verglichen werden sollen, müssen die jeweiligen absoluten Vorkommen mit einer geeigneten Normalisierungsgröße (sinnvoll sind z. B. alle Personalpronomina oder alle nominalen Wörter) relativiert (normalisiert) werden.

Die Kasus von Präpositionen sind im Deutschen relativ vielseitig, können wechseln und sind nicht immer einheitlich (z. B. sind *wegen*, *trotz*, *gemäß* und andere Präpositionen mit dem Genitiv und Dativ gebräuchlich). Es ist interessant, die Verteilung der präpositionalen Kasus im Deutschen in einem größeren Korpus zu analysieren. Hierbei fasst man die präpositionalen Kasus Nominativ (sofern existent), Genitiv, Dativ und Akkusativ als Varianten zu der Variable ›Präposition‹ bzw. ›präpositionaler Kasus‹ auf.

Ein Korpus, das Kasusinformationen enthält, ist das TIGER-Korpus. Es umfasst knapp eine Million Wortformen aus Zeitungstexten und eignet sich deshalb gut zur Quantifizierung grammatischer Aspekte in der deutschen Standardsprache.

Um auch hier den Lösungsweg nachvollziehen zu können, führen Sie die einzelnen Schritte der Anleitung aus.

Anleitung

- Öffnen Sie das Korpus »tiger2« im ANNIS-Suchinterface, in TIGERSearch oder auf der Webseite http://fnps.coli.uni-saarland.de:8080/query (zu mehr Informationen hierzu s. Kap. 3.1.2).
- Führen Sie vier Suchanfragen (für Nominativ, Genitiv, Dativ und Akkusativ) zu dem Kasus des Worts nach der Präposition aus. Zu bedenken ist hierbei, dass pränominale Genitive (›sächsische Genitive‹, z. B. *Peters Auto*) die Auswertung zugunsten des Genitivs verzerren können. Deshalb sollten Eigennamen, die in der Position des sächsischen Genitivs stehen, ausgeschlossen werden. Die Suchanfrage für den Nominativ ist in ANNIS:
  ```
  pos="APPR" . pos!="NE" _=_ morph=/.*Nom.*/
  ```
  Die Suchanfrage für die anderen beiden Instanzen lautet:
  ```
  [pos="APPR"] . [pos!="NE" & morph=/.*Nom.*/]
  ```
  Um die anderen Kasus abzufragen, nutzt man statt »Nom« die Kürzel »Gen«, »Dat« und »Acc«.
- Öffnen Sie LibreOffice (oder OpenOffice) Calc oder Microsoft Excel. Übertragen Sie die absoluten Zahlen pro Kasus in eine Tabelle analog zu der im vorigen Szenario: Die Kasus sollten untereinander in der linken Spalte, die Frequenzen zu den jeweiligen Kasus rechts daneben stehen.
- Erzeugen Sie wie im vorigen Szenario ein Säulendiagramm und Kreisdiagramm zur Visualisierung der Verteilung der einzelnen Häufigkeiten.
- Auch hier zeigen sich einige überraschende Fakten: Nominativ als präpositionaler Kasus ist (wider Erwarten) in den Daten vertreten und praktisch so häufig wie der Genitiv (sogar etwas häufiger). Es handelt

sich um Gleichsetzungskonstruktionen mit *als*, die relativ häufig im Korpus sind. Der Dativ ist deutlich häufiger (über zwanzig Mal) als der Genitiv und der Akkusativ ist ca. halb so häufig wie der Dativ und somit über zehnmal häufiger als der Genitiv.

Schreibungsvarianten im Mittelhochdeutschen: Historische Korpora können nicht nur die Veränderung im Sprachgebrauch, sondern auch die Heterogenität des Gebrauchs in bestimmten Zeitabschnitten aufzeigen. Anhand des Referenzkorpus Mittelhochdeutsch (http://www.linguistics.rub.de/rem/) kann man z. B. Varianten in der schriftlichen Repräsentation bestimmter Lexeme untersuchen. Um – als konkretes Anwendungsbeispiel – alle Schreibungsvarianten des Reflexivpronomens *sich* zu ermitteln, gehen Sie wie folgt vor.

Anleitung

Das Referenzkorpus Mittelhochdeutsch liegt durchsuchbar in einer Instanz des Suchwerkzeugs ANNIS vor: https://linguistics.rub.de/annis/annis3/REM.

- Öffnen Sie die Seite im Internetbrowser und markieren Sie unten links in der Auswahl sämtliche Subkorpora (es genügt für exemplarische Zwecke auch ein Teil der Korpusdaten).
- Geben Sie oben links den Suchausdruck `lemma="sich" _=_ tok_dipl` (ohne die umgebenden Anführungszeichen) ein. Die Suchvariable ›lemma‹ steht für die normalisierte Lemmaform, die Suchvariable ›tok_dipl‹ für die nicht normalisierte Wortform gemäß der Originalquelle.
- Wählen Sie die Funktion »More« > »Frequency Analysis« und warten Sie, bis sich das entsprechende Fenster geöffnet hat.
- Die mittige Tabelle zeigt die beiden Suchvariablen gemäß der Suchanfrage. Löschen Sie die erste (»lemma«; markieren Sie die Zeile und wählen Sie »Delete selected row(s)«), weil es um die Auswertung der anderen Suchvariable geht. Betätigen Sie »Perform frequency analysis«.
- Als Ergebnis erhalten Sie eine Übersicht sämtlicher Formen im Korpus, die für die abstrakte, normalisierte Form *sich* verwendet werden.

Arbeitsaufgaben

1. Erstellen Sie für das Korpus »Fuerstinnenkorrespondenz1.1« (Zugang: https://hu.berlin/annis-intro) analog zu den oben stehenden Anleitungen ein Variantenprofil, das alle Wortformen (Variable: »tok«) aufzeigt, deren Lemma (Variable: »lemma«) auf *-bar* endet. (Dies ist interessant, weil das Suffix *-bar* als produktiv für das aktuelle Deutsch gilt. Man kann hier sehen, ob dies im 16. – 18. Jh. auch schon so galt.)

2. Erstellen Sie für das Korpus »tiger2« (Zugang: https://hu.berlin/annis-intro) analog zu den oben stehenden Anleitungen ein Variantenprofil, das zeigt, als welche Wortart das Wort *an* besitzen kann. (Die relevanten Variablennamen hierfür sind »lemma« und »pos«.)

3. Erstellen Sie zu der Suchanfrage nach präpositionalem und postnominalem *wegen* in Aufgabe 1 des Kapitels 3.1.2.18 eine Übersicht zur Verteilung der Varianten
   a) präpositionales *wegen* mit komplexer Nominalphrase (Artikelwort und/oder Adjektiv) im Vorfeld
   b) präpositionales *wegen* mit Nomen und keinem anderen Wort (kein Artikelwort oder Adjektiv) im Vorfeld
   c) postpositionales *wegen* mit komplexer Nominalphrase (Artikelwort und/oder Adjektiv) im Vorfeld
   d) präpositionales *wegen* mit Nomen und keinem anderen Wort (kein Artikelwort oder Adjektiv) im Vorfeld

   im DeWaC-1-Korpus des CQP-Webinterfaces der Humboldt-Universität zu Berlin (Zugang über https://hu.berlin/cqp; Nutzername: CQP_Demo, Passwort: TestSuchen).

4.6.2 | Kookkurrenzen, Kollokationen und Kolligationen

Definition

- **Kookkurrenz** mindestens zweier sprachlicher Elemente meint ihr gemeinsames Auftreten in demselben sprachlichen Kontext.
- Von einer **Kollokation** spricht man dann, wenn Kookkurrenzen nicht zufällig sind. Dieser Fall liegt dann vor, wenn zwei (oder mehr) sprachliche Elemente signifikant häufiger als (statistisch) erwartet zusammen auftreten.
- Von einer **Kolligation** spricht man dann, wenn eine Kollokation grammatisch motiviert ist bzw. grammatisch klassifiziert werden kann. Die Kollokation eines Verbs und eines Nomens ist z. B. dann als Kolligation zu interpretieren, wenn das Nomen ein Objekt des Verbs ist.

Die drei Begrifflichkeiten lassen sich also taxonomisch verketten: Eine Kolligation ist eine spezifische Kollokation, eine Kollokation ist eine spezifische Kookkurrenz.

Es muss aber auch festgestellt werden, dass die Begriffe ›Kookkurrenz‹ und ›Kollokation‹ grammatisch nicht interpretiert sind und ›Kolligation‹ genau diese Interpretation liefert. Als Effekt hiervon sind ›Kookkurrenz‹ und ›Kollokation‹ stets mit dem Aspekt der positionellen Nähe verknüpft – es werden direkt nebeneinander stehende (adjazente) oder nah beieinander stehende Wörter, Lemmata oder Wortarten betrachtet. Eine Kolligationsanalyse lässt es zu, von diesen räumlichen Gesichtspunkten relativ zu abstrahieren. Vergleichen Sie z. B. Pronominaladverbien wie *darauf*

oder *daran*, die als Korrelate zu Nebensätzen mit *dass* stehen. Hierbei handelt es sich um das Zusammenauftreten von Elementen, die keinesfalls nebeneinander oder nahe beieinander stehen müssen (*Er möchte **darauf** nicht mehr unendlich lange warten müssen, **dass** ...*). Solche Strukturen lassen sich weder aufdecken noch überhaupt behandeln, wenn man zusammengehörige Einheiten nur unter der oben genannten Definition von Kollokation begreift. In diesem Sinne ist es sinnvoll, den Kolligationsbegriff weitestgehend von positionellen Aspekten zu lösen.

Ein vierter Begriff ›Kollostruktion‹ (engl. *collostruction*) wurde in Stefanowitsch/Gries (2003) etabliert, um gezielt die gegenseitige Anziehung zwischen Wörtern und grammatischen Strukturen (Konstruktionen) untersuchen zu können. Die Autoren nennen als Beispiel Verben, die die ›Intransitivkonstruktion‹ anziehen. Dies wird durch die Messung der Assoziationsstärke zwischen dem Auftreten von Dativ-Akkusativ-Abfolgen und Verben im selben Satz ermöglicht.

Kookkurrenzen bestehen also grundsätzlich, wo mindestens zwei sprachliche Elemente beieinander beobachtet werden. Beachten Sie das folgende Beispiel und stellen Sie sich vor, es entstammt einem großen Textkorpus.

(1) *Ich kann dir gern etwas Geld leihen und auch mein Auto zur Verfügung stellen .*

Jedes der 13 Wortpaare (man spricht linguistisch auch von adjazent stehenden Wörtern oder Wortformen) bildet eine Kookkurrenz. Auch das vierzehnte Zeichenpaar – *stellen* und das Satzbeendungszeichen – bildet eine Kookkurrenz, die in einem tokenisierten Korpus als gleichwertig zu den anderen Paaren berücksichtigt werden könnte. Nun erscheinen einige der Kookkurrenzen intuitiv betrachtet als zufällig (z. B. *dir-gern* oder *und-auch*), während andere Abfolgen auf der Beschaffenheit des Deutschen oder der Welt beruhen könnten. Zum Beispiel wissen wir, dass die Abfolge *mein-Werkzeug* eine sehr wahrscheinliche ist, weil der possessive Artikel häufig links neben einem Nomen steht. Die Folge *Geld-leihen* könnte aufgrund mindestens zweier Gründe eine »überproportional häufige« Abfolge sein, weil zum einen *leihen* ein Verb mit direktem Objekt ist und *Geld* als Nomen die Objektfunktion erfüllen kann, weil aber zum anderen *Geld* etwas ist, das häufig ver- bzw. geliehen wird, oder anders ausgedrückt, weil *leihen* etwas ist, das man häufig mit Geld macht. Zuletzt liegt die Vermutung auf der Hand, dass die Dreierkette *zur-Verfügung-stellen* nicht zufällig, sondern sprachlich bedingt auftritt, weil die drei Elemente eine Mehrworteinheit bilden.

Bei diesen drei letztgenannten Beispielen handelt es sich also um (zunächst mutmaßliche) Kollokationen, die aufgrund ganz unterschiedlicher Ursachen existieren: aufgrund der Syntax des Deutschen, aufgrund der Zusammenhänge der sozialen Welt, aufgrund lexikalischer Faktoren. Bei der Untersuchung der genannten Einheiten müsste man zunächst beweisen, dass die jeweiligen Paare tatsächlich eine Kollokation bilden; anschließend kann man die Kollokation grammatisch interpretieren und somit die Art der Kolligation beschreiben.

Das Ermitteln von Kollokationen findet statt über den Abgleich des erwarteten Zusammenauftretens zweier kookkurrierender Elemente und deren tatsächliches Zusammenauftreten. Je mehr das tatsächliche Zusammenauftreten das erwartete übersteigt, umso mehr gelten die beiden Elemente als assoziiert. Mathematisch ausgedrückt: Liegt der Quotient aus dem beobachteten gemeinsamen Auftreten zweier Elemente x und y und dem erwarteten gemeinsamen Auftreten über eins, handelt es sich um eine positive Assoziation. Man kann dann für die ausgewerteten Daten sagen, dass x und y eine Kollokation bilden. Ist der Quotient unter eins, so handelt es sich um eine negative Assoziation und somit um keine Kollokation innerhalb der analysierten Daten.

Die konkreten Formeln, die die Assoziation errechnen, nennt man Assoziationsmaße. Sie nähern sich dem erwarteten Auftreten zweier Elemente bzw. der Relation zwischen tatsächlichem und erwartetem Auftreten auf verschiedene Weise, weil der tatsächliche Erwartungswert in aller Regel schwer ermittelt werden kann (s. u.). In der Regel geben die verwendeten Maße logarithmierte Werte aus, weil so alle primären Rechenergebnisse über eins und somit positive Assoziationen einen positiven Wert erhalten und alle primären Ergebnisse zwischen null und eins einen negativen Assoziationswert erhalten.

Ein häufig verwendetes Assoziationsmaß heißt Mutual Information (gegenseitige Information). Ebenso wird (z. B. im DWDS-Wortprofil, http://www.dwds.de/wp?q=) der Dice-Koeffizient verwendet, um Kollokationen zu ermitteln, auch wenn hierbei wesentliche relevante Faktoren fehlen und es sich nicht um ein Assoziationsmaß, sondern ein Ähnlichkeitsmaß handelt, das lediglich die beobachteten Frequenzen des alleinigen und des gemeinsamen Auftretens zweier Elemente in Beziehung setzt. Für die Berechnung eines Assoziationswertes für zwei Elemente x und y innerhalb eines Korpus nach dem Dice-Koeffizienten benötigt man die Anzahl der Kookkurrenzen von x und y sowie die Vorkommen von x und y im Korpus. Für die Berechnung des Mutual-Information-Maßes muss zusätzlich eine Bezugsgröße in die Berechnung einfließen. Meistens wird die Korpusgröße in Token herangezogen, die Angemessenheit dieses Vorgehens ist jedoch äußerst umstritten.

In dreifacher Hinsicht sei hier zur Vorsicht gemahnt: Erstens dürfen die Ergebnisse solcher Berechnungen nicht mit Signifikanzen verwechselt werden, denn auch starke Assoziationen müssen aufgrund der Größe der analysierten Daten und der Zusammensetzung der Daten nicht signifikant, sondern können zufällig sein. Je seltener die gemessenen Elemente, umso größer muss die für eine zuverlässige Berechnung erforderliche Datenmenge sein. Zweitens muss die Definition von Kookkurrenz je nach dem Ziel der Messung (der ihr zugrunde liegenden Fragestellung) kritisch hinterfragt werden. Drittens müssen die genannten Maße hinsichtlich ihres konkreten Anwendungsbereichs manipuliert bzw. spezifiziert werden – je nach Korpusgröße und der relativen Häufigkeit der beobachteten Phänomene müssen gewisse Faktoren in die Berechnung einfließen, damit sich ein vernünftig interpretierbarer Ergebniswert ergibt. Ansonsten können Ergebnisse von Assoziationsberechnungen nicht für sich, sondern immer nur im Vergleich zu anderen Elementpaaren und deren indi-

viduellem Berechnungsergebnis interpretiert werden. Für die standardmäßige Berechnung von Kollokationen mit dem Maß »mutual information« in der Linguistik vgl. Biber/Jones 2009, S. 1295 f. Für eine kritische Anwendung vgl. das folgende Beispiel.

Anwendungsbeispiel: An dem oben eingeführten Beispiel der Hypothese, dass *Geld* und *leihen* eine Kollokation bilden, sollen diese Probleme genauer besprochen werden. In dem Beispielsatz oben stellen sie eine Kookkurrenz dar. Um zu überprüfen, ob sie eine Kollokation darstellen, muss man an einem geeigneten Korpus als Datengrundlage überprüfen, ob das gemeinsame Auftreten beider Wörter häufiger ist als ihr erwartetes gemeinsames Auftreten. Hierbei ergibt sich ein fundamentales Problem: Wie errechnet man das erwartete gemeinsame Auftreten solcher zwei Wörter?

Messen wir zunächst ein bestimmtes Korpus auf die Kookkurrenzen zu den beiden Wörtern aus: Wir nehmen das Korpus »DeWaC 1« im CQP-Interface der Humboldt-Universität zu Berlin (zu Zugangs- und Nutzungshinweisen s. Kap. 3.1.2). Dieses Korpus besteht aus deutschsprachigen Internetdaten und ist mit 270 Millionen Token relativ groß. Wir zählen mit der Suchanfrage

```
[lemma="Geld"][lemma="leihen"] | [lemma="leihen"][lemma="Geld"]
```

unmittelbare Kookkurrenzen der beiden besagten Wörter im DeWaC-1-Korpus. Wir wählen als Bezugsgröße Lemmata, weil der Beugungsstatus der beiden Wörter für die Frage nicht relevant erscheint. Es handelt sich um 68 Fälle. Hierbei ergibt sich das erste methodische Problem: Man sieht, dass die erste Abfolge mit 66 Vorkommen viel höher ist als die zweite Abfolge mit nur zwei Vorkommen. Dies liegt insofern an der deutschen Syntax, als dass das Auftreten von *Geld* unmittelbar nach dem Verb im Normalfall wegen anderer Satzglieder, die im Hauptsatz zwischen dem Verb und dem direkten Objekt stehen, verhindert wird, sofern das Verb in der linken Satzklammer steht. Dabei steht *Geld* grammatisch gesehen genauso mit dem Verb *leihen* zusammen, wenn es später im Satz als Objekt erscheint, wie wenn es im Nebensatz oder in einem Satz mit komplexem Prädikat unmittelbar neben dem Verb *leihen* steht. Dies zeigt ein wesentliches Problem von Kollokationen: Linguistisch gesehen, ist der zentrale Positionsaspekt der Kollokationen nicht aussagekräftig. Eine Lösung ist, den Kollokationsbegriff auf den Satzkontext auszuweiten und alle Vorkommen von *Geld* und *leihen* innerhalb von Sequenzen von Satzzeichen zu suchen. Die entsprechende Suchanfrage in CQP lautet

```
[lemma="Geld"][pos!="\$."]*[lemma="leihen"] | [lemma="leihen"][pos!="\$."]*[lemma="Geld"]
```

und liefert für das ausgewählte Korpus 173 Treffer.

Zu den übrigen Frequenzen: Das Verb *leihen* zählt 1452 Fälle im Korpus, das Nomen *Geld* 56379. Daran gemessen, scheint das gemeinsame Auftreten von nur 68 bzw. 173 Mal relativ selten zu sein. Aber um dies

rechnerisch zu erfassen, muss man wissen, wie viele mögliche Kombinationen pro Auftreten eines Worts vorliegen und wie wahrscheinlich die Kookkurrenz zweier bestimmter Wörter ist. Gemessen an dieser Wahrscheinlichkeit kann man die Stärke der Kollokation errechnen.

In diesem Sinne wird das ›Mutual Information‹-Maß immer errechnet durch den Quotienten aus der beobachteten Kookkurrenz und der erwarteten Wahrscheinlichkeit der Kookkurrenz zweier Einheiten. Alle Ergebnisse mit einem Wert größer als eins besitzen eine positive Assoziation und sind somit Kollokationen, alle Werte kleiner eins und größer null besitzen eine negative Assoziation und sind somit keine Kollokationen.

Werte für die beobachtete Kookkurrenz liegen bereits vor; die Wahrscheinlichkeit für das gemeinsame auftreten muss anhand der Häufigkeiten der getesteten Kandidaten und mindestens einer dritten Größe errechnet werden. Diese dritte Größe ist ein fundamentales Problem aller Assoziationsberechnungen für sprachliche Elemente: Sie ist in der Forschungstradition schlicht die Korpusgröße, angegeben in Wort- oder Tokenhäufigkeiten. Im Fall des zur Berechnung genutzten DeWaC-1-Korpus herangezogenen handelt es sich um 268.849.871 Token. Die erwartete Häufigkeit beträgt somit gerundet 0,3 (das Ergebnis aus 1452, multipliziert mit 56379, geteilt durch 268.849.871). Der resultierende Wert für MI (Mutual Information) ist für den Wert 68 (für das adjazente Auftreten) 223,3, für den Wert 173 (gemeinsames Auftreten im Satz) gerundet 568,2, also in beiden Fällen extrem hoch. Diese Werte sind für sich nicht interpretierbar, weil sie hier nicht die Wahrscheinlichkeit der Kookkurrenz an der tatsächlichen Kookkurrenz gemessen wurde. Wenn wir weitere Verb-Nomen Paare auf die besagte Weise vermessen, lassen sich die jeweiligen Ergebnisse miteinander vergleichen, und somit kann relativ gesehen ermittelt werden, welche Paare sich verhältnismäßig anziehen und welche dies weniger tun. Wenn andere Kategorienpaare, z. B. Nomina und Artikelwörter, auf ihre Kollokationsstärke hin gemessen werden, dürfen diese Ergebnisse in keinem Fall mit denen der Verb-Nomen-Paare in Beziehung gesetzt werden, weil die Bezugsgröße nicht jeweils die Tokenzahl des Korpus sein dürfte, sondern korrekterweise jeweils die Anzahl der möglichen Kookkurrenzpartner sein müsste (bei dem Verb-Nomen-Fall könnte man näherungsmäßig für ein bestimmtes Verb die Anzahl der verschiedenen Nomen-Lemmata, bei dem Nomen-Artikel-Fall die Anzahl der verschiedenen Artikel-Lemmata im Korpus nehmen).

Arbeitsaufgabe

Verfolgen Sie das Auswertungszenario wie beschrieben weiter und ermitteln Sie anhand der »DeWaC 1«-Korpusdaten vergleichend zu dem Paar *leihen* und *Geld* MI-Werte (Mutual Information) für *leihen* und *Ohr*, *leihen* und *Bleistift* sowie *schenken* und *Aufmerksamkeit*.

4.7 | Korrelationen

Definition

Eine Korrelation bezeichnet in der Linguistik die Abhängigkeit zweier (oder mehrerer) quantifizierbarer Merkmale. Häufig wird aus Beobachtungen abgeleitet, das ein Merkmal y verstärkt (also häufiger) auftritt, wenn ein Merkmal x verstärkt auftritt. Hierbei spricht man von einer positiven Korrelation. Eine negative Korrelation bezeichnet die umgekehrt gepolte Abhängigkeit.

Beispiele für Annahmen aus der Linguistik, die Korrelationen darstellen, sind z. B. folgende:

- Die relative Anzahl der Nomina pro Text korreliert positiv mit der Satzlänge (je mehr Nomen, desto länger die Sätze), während die relative Anzahl der Verben pro Text negativ mit der Satzlänge korreliert.
- Je (konzeptionell) mündlicher die Sprache, umso kürzer sind die Sätze.
- Die Länge der Wörter und Sätze korreliert positiv mit der Sprachkompetenz von Lernenden des Deutschen als Fremdsprache.

An den Beispielen zeigt sich gut, dass Korrelationen sowohl genutzt werden können, um eine bestimmte Varietät abzubilden, aber auch ideal zum Vergleich verschiedener Varietäten verwendet werden können. Zum Beispiel kann der Nachweis, dass Schriftlichkeit positiv mit verschiedenen Komplexitätsmaßen korreliert, dazu dienen, mündliche und schriftliche Register miteinander zu vergleichen. ~~Eine solche Korrelation kann aber auch als Merkmal der Dynamik innerhalb derselben Standardvarietät interpretiert werden.~~

Der Nachweis über eine Korrelation hat zunächst nichts mit der Art des Zusammenhangs zu tun. Hierfür lassen sich aus dem Alltag diverse Beispiele finden. Wenn z. B. ein positiver Zusammenhang zwischen Zuckerkonsum und Karies oder der Grad der Kälte und der Anzahl der Parkbesucher besteht, so liegt eine kausale Relation zwischen den Variablen nahe (aufgrund zunehmender Kälte gehen die Menschen weniger in den Park). Doch lassen sich auch Korrelationen in der Welt ausmachen, die nicht unmittelbar in einem Kausalzusammenhang stehen. Widersprüchlicherweise werden solche Fälle als Scheinkorrelation bezeichnet, obwohl ja eine Korrelation tatsächlich besteht. Als Beispiel wird immer wieder angeführt, dass die Anzahl der Störche mit der Anzahl der Geburten korreliere. Dass dies nicht unmittelbar kausal bedingt ist, liegt auf der Hand, doch man kann die Korrelation durch die Drittvariable der ›Ländlichkeit‹ erklären: In ländlichen Gebieten ist die Anzahl der Störche und die relative Geburtenrate höher als in weniger ländlichen Gebieten. Es lassen sich auch Korrelationen zwischen Variablen messen, zwischen denen überhaupt kein kausaler Zusammenhang auszumachen ist. So sammelt Tyler Vigen auf seiner Webseite http://www.tylervigen.com/spurious-correlations Beispiele für aberwitzige Korrelationen wie z. B. den durch-

schnittlichen Pro-Kopf-Konsum von Margarine und der Scheidungsrate pro Jahr im US-Bundesstaat Maine. In diesem Sinne darf auch in der Linguistik eine gemessene Korrelation nicht automatisch als Kausalzusammenhang interpretiert werden.

Die Korrelation zweier Variablen muss über die Messung der Variablen in verschiedenen Datenmengen erfolgen, in denen ihr Auftreten variiert. Diese Datenmengen können einzelne Korpora sein oder auch Ausschnitte aus demselben Korpus, z. B. die Texte, aus denen das Korpus zusammengesetzt ist. Stellen Sie sich vor, Sie messen die relative Häufigkeit von Verben; sie beträgt in einem Korpus (im TIGER-Zeitungskorpus) pro 100 Token 12,0 Vorkommen und in einem anderen pro 100 Token 16,2 Vorkommen (im Korpus »Fuerstinnenkorrespondenz« im ANNIS-Suchinterface der Humboldt-Universität zu Berlin). Wenn man annimmt, dass die relative Anzahl der Verben negativ mit der relativen Anzahl der Nomina korreliert, so ist zu erwarten, dass im TIGER-Korpus die relative Anzahl der Nomina höher ist als im Vergleichskorpus »Fuerstinnenkorrespondenz«. Tatsächlich beträgt die relative Nomenzahl pro 100 Token im TIGER-Korpus 26,6 und in »Fuerstinnenkorrespondenz« 17,8. Wie man Korrelationen etwas präziser misst, wird in dem folgenden Datenbeispiel erläutert.

Datenbeispiel zur Messung von Korrelation: Wir greifen das Datenbeispiel zur (negativen) Korrelation von Nomina und Verben auf und nehmen noch zwei Vergleichsdatenpunkte (Korpora) hinzu: das »pcc2.1«-Korpus und das Korpus »Parlamentsreden_Deutscher_Bundestag«, beide online verfügbar über das in Kapitel 3.1.2 vorgestellte ANNIS-Suchinterface der Humboldt-Universität zu Berlin. Tabellarisch lässt sich die Gegenüberstellung der relativen Verb- und Nomenfrequenz in den vier Korpora wie in Tab. 4.7 darstellen.

Die Tendenz einer Korrelation lässt sich am besten beobachten, wenn man Daten wie die in der Tabelle gezeigten mit einem Streudiagramm (Punktdiagramm) visualisiert. Hierbei werden die einzelnen Datenpaare pro Korpus auf ein zweidimensionales Koordinatensystem abgetragen, wobei die beiden Achsen für die beiden Variablen stehen (im aktuellen Beispiel bildet die x-Achse die Frequenz der Verben und die y-Achse die Frequenz der Nomina ab). Daraus ergibt sich das Bild gemäß Abb. 4.20.

Wie man eine Abbildung wie Abb. 4.20 anhand der oben gezeigten Daten erzeugt, erfahren Sie am Ende des Kapitels in der Anleitung.

Man sieht, dass eine negative Korrelation bezogen auf alle Datenpunkte vorliegt: Je höher die Anzahl der Verben steigt, desto geringer

Korpus	Variable 1: Verben (pro 100 Token)	Variable 2: Nomina (pro 100 Token)
TIGER	12,0	26,6
Fuerstinnenkorrespondenz	16,2	17,8
PCC	13,7	20,7
Parlamentsreden	12,5	25,0

Tab. 4.7: Relative Häufigkeit von Verben und Nomina in den Korpora TIGER, Fuerstinnenkorrespondenz, PCC und Parlamentsreden

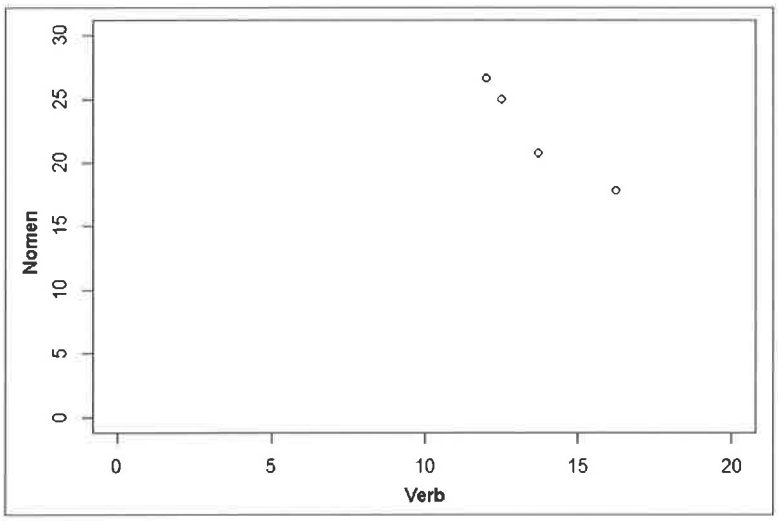


Abb. 4.20: Visualisierung der Korrelation von Verben und Nomina in den vier untersuchten Korpora

wird die Anzahl der Nomina und umgekehrt. Die Korrelation lässt sich aber nicht nur deskriptiv, sondern auch inferentiell-statistisch bearbeiten: Es existieren verschiedene statistische Methoden zur Bemessung der Korrelationsstärke (dies sind deskriptive Verfahren) und des Signifikanzniveaus der Korrelation (dies sind inferentielle Verfahren). Die Maße für die Stärke nennen sich Korrelationskoeffizienten und liefern Werte von –1 (maximale negative Korrelation) bis 1 (maximale positive Korrelation) liefern. Schwache bis nicht als solche zu interpretierende Korrelationen werden durch einen Wert um null herum ausgedrückt. Das bekannteste Maß für die Korrelation ist der Pearson-Korrelationskoeffizient r, bei welchem die Werte wie die oben angegebenen sowie die Mittelwerte aus den beiden Zahlenreihen und die Anzahl der Messpunkte miteinander verrechnet werden. Dieses Maß lässt sich in den Programmen R (bzw. der Oberfläche RStudio), LibreOffice Calc oder Microsoft Excel berechnen und liefert für die gegebenen Daten einen Wert r = –0,96, also den Wert für eine relativ starke negative Korrelation. Das Signifikanzniveau, welches sich zu den Daten berechnen lässt, beträgt gerundet 0,039 und ist somit gemäß dem Schwellenwert von 0,05 im signifikanten Bereich.

Den folgenden Verarbeitungsschritten liegen die oben aufgeführten Korpusdaten zugrunde, die sich aus den relativen Häufigkeiten der Wortkategorien Verb und Nomen in den Korpora »tiger2«, »Fuerstinnenkorrespondenz1.1«, »pcc2.1« und »Parlamentsreden_Deutscher_Bundestag« (alle verfügbar im ANNIS-Suchinterface unter der Adresse https://hu.berlin/annis-intro) ergeben. Verfahren Sie zur Ermittlung dieser Frequenzen sowie zu ihrer Weiterverarbeitung wie folgt:

Anleitung

- Stellen Sie Suchanfragen nach allen Verben und allen Nomina in den einzelnen Korpora. Beachten Sie dabei, dass die Suchvariable für Wortarten im Parlamentsredenkorpus »POS« heißt, in den übrigen Korpora »pos«. Alle Korpora verwenden das Wortartentagset STTS.
  - Schreiben Sie die absoluten Frequenzen pro Korpus heraus.
  - Normalisieren Sie diese Zahlen an den Tokenzahlen der Korpora, die in der Korpusübersicht (»Tokens«, links unten im ANNIS-Suchinterface) stehen.
  - Ordnen Sie die normalisierten Werte an, wie oben in der Tabelle gezeigt.
- Für eine Weiterverarbeitung in LibreOffice (oder OpenOffice) Calc oder Microsoft Excel kopieren Sie die Übersicht in diese Programme, so dass die Zahlen im Bereich der Zellen von B2 (links oben) bis C5 (rechts unten) stehen.
  - Markieren Sie die Zahlenwerte und wählen Sie »Einfügen« > »Diagramm« > »Streudiagramm« bzw. »Einfügen« > »Punkt« (Excel). Sie erhalten das gewünschte Diagrammformat.
  - Gehen Sie in eine leere Zelle unterhalb der Daten und geben Sie die Formel `=KORREL(B2:B5;C2:C5)` ein. (Die durch die Doppelpunktausdrücke definierten Bereiche kann man durch das Markieren der entsprechenden vertikalen Zellenreihen eingeben und modifizieren.) Sie erhalten so einen Wert für den Pearson-Korrelationskoeffizienten r.
- Für eine Weiterverarbeitung in RStudio müssen die Daten in das folgende Format gebracht werden:

```
Verb<-c(12.0,16.2,13.7,12.5)
Nomen<-c(26.6,17.8,20.7,25.0)
```

  - Kopieren Sie diese Daten nacheinander in die R-Studio-Console (unten links) und bestätigen Sie jeweils mit Enter.
  - Geben Sie den folgenden Befehl ein:

```
plot(Verb,Nomen,xlim=c(0,20),ylim=c(0,30))
```

    Hierdurch erhalten Sie die Visualisierung (man sagt: »den Plot der Daten« bzw. »Die Daten werden geplottet«). Die Funktionen »xlim« sowie »ylim« legen die Skalierung der Achsen fest.
  - Geben Sie den folgenden Befehl ein:

```
cor.test(Verb,Nomen,method="pearson")
```

    Hiermit führen Sie die Berechnung des Pearson-Korrelationskoeffizienten aus und erhalten zudem einen Wert für die Signifikanz der Berechnung (p-Wert).
- Sie können unter der Internetadresse https://bit.ly/2Tns3cc auch die vier RStudio-Befehle in RStudio öffnen und mittels STRG-Enter-Kombination Befehl für Befehl ausführen (zuvor muss der Cursor in die Zeile des Befehls gesetzt werden).

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_4

Hinweis: Abhängigkeiten zwischen mehr als zwei Variablen lassen sich unter dem Begriff ›multivariate Verfahren‹ (auch: mehrfaktorielle Methoden oder Analysen) zusammenfassen, unter die Clusteranalysen und andere Methoden fallen (eine Sammlung solcher Verfahren findet sich in Gries 2008, S. 241 f., eine Vorstellung des Konzepts von gemischten Effekten bzw. ›mixed-effects‹ in Gries 2013, S. 333 f.).

5 Serviceteil

5.1 Internetlinks zu frei verfügbaren Korpusressourcen
5.2 Zitierte Literatur
5.3 Sachregister

5.1 | Internetlinks zu frei verfügbaren Korpusressourcen

Die folgenden Übersichten fassen, alphabetisch geordnet, im Buch vorgestellte Werkzeuge zur Korpusaufbereitung, zur Korpussuche bzw. -auswertung sowie einzelne Korpora zusammen.

Vorgestellte Werkzeuge zur Korpusaufbereitung

Hinweis zum Verständnis der Einteilungen: Ob die Annotationen **manuell** oder **automatisch** erstellt werden, gibt die entsprechende Spalte an. Es gibt Ressourcen, die **online** per Browser zugänglich sind, die herunterladbar und **lokal** nutzbar und/oder die auf einem (eigenen) **Server** zu installieren sind. Wenn ein Werkzeug nicht als ›lokal‹ ausgewiesen ist, kann es nicht auf dem eigenen Computer installiert werden.

Name der Ressource	Annotationstypen	manuell/automatisch	Erwähnung auf S. ...	Verfügbarkeit (online/lokal/Server)
Arborator	Syntax/Dependenzen	manuell	■■	online, Server
Homepage: Annotation:	https://arborator.ilpga.fr/ https://arborator.ilpga.fr/q.cgi (möglichst Chrome-Browser verwenden)			
CorZu	Koreferenz	automatisch	■■	online, lokal, Server
Homepage: Download: Online-Demo:	https://bit.ly/2TWkFJv https://github.com/dtuggener/CorZu https://pub.cl.uzh.ch/demo/corzu/			
EXMARaLDA	Spannen; beliebige Kategorien	manuell	■■ (Annotation), ■■ (Transkription)	lokal
Homepage: Download:	https://exmaralda.org/de/ https://exmaralda.org/de/offizielle-version/			
MaltParser	Syntax/Dependenzen	automatisch	■■	lokal, Server
Homepage: Download: Online-Demo:	http://www.maltparser.org/ http://www.maltparser.org/download.html http://de.sempar.ims.uni-stuttgart.de/			

H. Hirschmann, *Korpuslinguistik*, https://doi.org/10.1007/978-3-476-05493-7_5

Name der Ressource	Annotations-typen	manuell/ auto-matisch	Erwäh-nung auf S. ...	Verfügbarkeit (online/lokal/Server)
MMAX2	Koreferenz	manuell	■■	lokal
Qickstart-Gui-de: Download:	http://mmax2.net/mmax2quickstart.pdf https://sourceforge.net/projects/mmax2/			
ParZu	Syntax/Depen-denzen	auto-matisch	■■	lokal, Server
Download: Online-Demo:	https://github.com/rsennrich/ParZu https://pub.cl.uzh.ch/demo/parzu/			
rsttool	RST-Bäume	manuell	■■, ■■ (An-leitung)	lokal
Homepage RST: Homepage Tool: Download:	http://www.sfu.ca/rst/ http://www.wagsoft.com/RSTTool/ http://www.wagsoft.com/RSTTool/section2.html			
TreeTagger	Wortarten, Lem-mata	auto-matisch	■■, ■■ (An-leitung f. TagAnt)	lokal, Server
Homepage/ Download: Online-In-stanz: Download Ta-gAnt:	http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ https://copa-trad.ufsc.br/#tree-tagger-cloud http://www.laurenceanthony.net/software/tagant/			
WebAnno	diverse; Syntax, Informations-struktur	manuell	■■	lokal, Server
Homepage: Download:	https://webanno.github.io/webanno/ https://webanno.github.io/webanno/downloads/			
WebLicht	diverse	auto-matisch	■■	online
Homepage: Service:	https://bit.ly/2OhXe7V https://weblicht.sfs.uni-tuebingen.de/weblicht/			

Vorgestellte Werkzeuge zur Korpussuche

Name der Ressource	beschrieben ab S.	Verfügbarkeit
ANNIS	■■, ■■	online, eigener Server, lokal
Software: HU-Instanz allgemein: HU-Instanz Falko (DaF): HU-Instanz historisch: HU-Instanz Tutorial: MERLIN (Lernersprache): Georgetown (diverse): Referenzkorpus Mittelhochdeutsch: Referenzkorpus Mittelniederdeutsch:	http://corpus-tools.org/annis/ https://hu.berlin/annis https://hu.berlin/annis-falko https://hu.berlin/annis-ddd https://hu.berlin/annis-intro http://www.merlin-platform.eu/ https://bit.ly/2JyfwTQ https://bit.ly/2TjTT9n https://bit.ly/2Ojp43H	
AntConc	■■	lokal
Homepage/Download:	https://bit.ly/1MeMh0f	
COSMAS	■■	online
Zugang (mit Registrierung):	https://bit.ly/1MJcS51	
CQP	■■	online, Server
Homepage/Download: HU-Webinterface: TU-Dresden-Webinterface: Uni-Erlangen-Webinterface: Webinterface d. Uni Saarland:	http://cwb.sourceforge.net/ https://hu.berlin/cqp http://linguistik.zih.tu-dresden.de/corpus/ https://bit.ly/2FqzpId https://bit.ly/2U6MSNw	
DGD (Datenbank gespr. Deutsch)	■■	online
Zugang (mit Registrierung):	https://dgd.ids-mannheim.de/	
DTA und DWDS	■■	online
DTA – Homepage und Zugang: DWDS – Homepage und Zugang:	http://www.deutschestextarchiv.de/ http://www.dwds.de	
NoSketch Engine	■■	online, Server
Homepage/Download: COW-Korpora-Interface (Registrierung): Interface CLARIN (Slowenien):	https://nlp.fi.muni.cz/trac/noske http://corporafromtheweb.org/ https://bit.ly/2TOAYbO	
TIGERSearch	■■	lokal (u. online)
Homepage/Download: Uni-Saarland-online-Interface:	https://bit.ly/2TkhTsV http://fnps.coli.uni-saarland.de:8080/query	
TüNDRA	■■	online
Auswahl u. Zugang (über Instituts-login):	https://bit.ly/2W9cbMm	

Frei zugängliche, im Buch behandelte Korpora

Name der Ressource	Varietät	Verfügbarkeit	Zugang über
BeMaTaC	gesprochene Dialoge L2 u. L1	online, Download	ANNIS (HU-Instanz)
Beschreibung und Download: Zugang zur Suche (L2-Korpus): Zugang zur Suche (L1-Korpus):	https://hu.berlin/bematac https://bit.ly/2FkXyyK (Suche nach Wort »hast«) https://bit.ly/2CyNGky (Suche nach Wort »hast«)		
CHILDES	Kindersprache	online, Download	Talkbank
Übersichtsseite: Daten (Deutsch): Suche (deutsche Daten):	https://childes.talkbank.org/ https://childes.talkbank.org/access/German/ https://bit.ly/2CraKBF		
DECOW	Internetsprache	online	NoSketch Engine
Beschreibung: Korpuszugang:	https://bit.ly/2ulN8tK http://www.webcorpora.org/		
DeReKo	gemischt	online	COSMAS
Beschreibung: Korpuszugang (Registrierung):	http://www1.ids-mannheim.de/kl/projekte/korpora https://bit.ly/1MJcS51		
DeWaC	Internetsprache	online, Download	CQP (HU Berlin), NoSketch Engine (CLARIN Slowenien)
Beschreibung: Zusammenfassung in NoSketch E.: Korpuszugang (NoSketch Engine): Korpuszugang (CQP HU Berlin):	https://bit.ly/2FowjU7 https://bit.ly/2CwouLj https://bit.ly/2TOAYbO https://hu.berlin/cqp (Logindaten S. ■■)		
Falko	Lernersprache DaF	online, Download	ANNIS (HU-Instanz DaF)
Beschreibung: ANNIS-Interface (HU):	https://hu.berlin/falko https://hu.berlin/annis-falko		
FOLK	gesprochenes Deutsch	online	DGD
Beschreibung: Zugang (Registrierung erforderl.):	http://agd.ids-mannheim.de/folk.shtml https://dgd.ids-mannheim.de/		
Fürstinnenkorrespondenzen	hist. Deutsch	online	ANNIS (HU Berlin)
Beschreibung: Korpuszugang (ANNIS HU)	https://bit.ly/2U5nzeM https://bit.ly/2UNIW1z (Suche nach Wort »hast«)		

Name der Ressource	Varietät	Verfügbarkeit	Zugang über
GeWiss	gesprochenes Deutsch, wissenschaftl.	online	DGD
Beschreibung: Zugang (Registrierung erforderl.):	https://bit.ly/2Hyx00A https://dgd.ids-mannheim.de/		
MERLIN	Lernersprache DaF	online	ANNIS-MERLIN
Beschreibung: Korpuszugang:	https://bit.ly/2WeJy0f https://merlin-platform.eu/		
Referenzkorpus Altdeutsch	hist. Deutsch	online	ANNIS (HU Berlin)
Beschreibung: Zugang (ANNIS HU Berlin):	http://www.deutschdiachrondigital.de/manual/ https://hu.berlin/annis-ddd		
Referenzkorpus Mittelhochdeutsch	hist. Deutsch	online	ANNIS (Ruhr-Universität Bochum)
Beschreibung: Vereinfachte Suchmaske: ANNIS-Interface (RUB):	http://www.linguistics.rub.de/rem/ https://bit.ly/2WdhEll https://linguistics.rub.de/annis/annis3/REM/		
Referenzkorpus Mittelniederdeutsch	hist. Deutsch	online	ANNIS (Universität Hamburg)
Beschreibung: Vereinfachte Suchmaske: ANNIS-Interface (Uni HH):	http://www.slm.uni-hamburg.de/ren/korpus/ https://bit.ly/2TOdCmK http://annis.corpora.uni-hamburg.de:8080/gui/ren		
Parlamentsreden Deutscher Bundestag	Plenar-protokolle	online	CQP (HU Berlin), ANNIS (HU Berlin)
Korpuszugang (CQP HU Berlin): Korpuszugang (ANNIS HU Berlin):	https://hu.berlin/cqp (Logindaten S. ■■) https://bit.ly/2FeNqaE (Suche nach Wort »hast«)		
RIDGES	hist. Deutsch	online, Download	ANNIS (HU Berlin)
Projektseite: Dokumentation: Download: Korpuszugang:	https://hu.berlin/ridges https://hu.berlin/ridges-doku https://hu.berlin/ridges-download https://hu.berlin/ridges-korpus (Suche nach Wort »Beispiel«)		
TIGER	Zeitungssprache	lokal, online	TIGERSearch, Webinterface (Uni Saarland), ANNIS (HU Berlin)

Name der Ressource	Varietät	Verfügbarkeit	Zugang über
Beschreibung: TIGERSearch & TIGERRegistry: Download Korpusdaten: Korpuszugang (Uni Saarland): Korpuszugang (HU Berlin):	https://bit.ly/2Y9eOjf https://bit.ly/2uqyUrj (unten auf der Seite) https://bit.ly/2Whhrh3 http://fnps.coli.uni-saarland.de:8080/query https://bit.ly/2Jyul8T (Suche nach Genitivobjekten)		
TüBa-D/Z	Zeitungs-sprache	lokal, online	TIGERSearch, TüNDRA
Beschreibung und Datenbestellung: Korpuszugang über TüNDRA:	https://bit.ly/2JuMjJt (Hinweise zum Bezug der Korpusdaten finden sich ganz unten auf der Seite.) https://bit.ly/2UP1oqr (Zugang über Institutslogin; TüBa-D/Z 10 ist bereits ausgewählt.)		

5.2 | Zitierte Literatur

Abney, Steven (1991): Parsing by Chunks. In: Berwick, Robert/Abney, Steven/Tenny, Carol (Hg.): *Principle-Based Parsing*. Dordrecht: Kluwer.

Albert, Stefanie/Anderssen, Jan/Bader, Regine/Becker, Stephanie/Bracht, Tobias/Brants, Sabine/Brants, Thorsten/Demberg, Vera/Dipper, Stefanie/Eisenberg, Peter/Hansen, Silvia/Hirschmann, Hagen/Janitzek, Juliane/Kirstein, Carolin/Langner, Robert/Michelbacher, Lukas/Plaehn, Oliver/Preis, Cordula/Pußel, Marcus/Rower, Marco/Schrader, Bettina/Schwartz, Anne/Smith, George/Uszkoreit, Hans (2003): *TIGER Annotationsschema*. Technischer Bericht. Universität Potsdam/Universität des Saarlandes/Universität Stuttgart. (http://www.linguistics.ruhr-uni-bochum.de/~dipper/pub/tiger_annot.pdf)

Anthony 2015

Auer, Peter (2010): Zum Segmentierungsproblem in der Gesprochenen Sprache. In: *InLiSt* (*Interaction and Linguistic Structures*) 49. (http://www.inlist.uni-bayreuth.de/issues/49/InList49.pdf)

Baayen, Harald R. (2008): *Analyzing Linguistic Data. A Practical Introduction to Statistics Using R*. Cambridge: Cambridge University Press.

Belz, Malte (2014): *Richtlinien zur Annotation von Reparaturen in BeMaTaC*. Technischer Bericht. Humboldt-Universität zu Berlin. (https://hu.berlin/bematac-guidelines)

Benikova, Darina/Yimam, Seid Muhie/Santhanam, Prabhakaran/Biemann, Chris (2015): GermaNER: Free Open German Named Entity Recognition Tool. In: *Proceedings of International Conference of the German Society for Computational Linguistics and Language Technology* (*GSCL-2015*). Universität Duisburg-Essen.

Biber, Douglas (1988): *Variation across Speech and Writing*. Cambridge: Cambridge University Press.

Biber, Douglas/Conrad, Susan (2009): *Register, Genre, and Style*. Cambridge: Cambridge University Press.

Biber, Douglas/Jones, James K. (2009): Quantitative Methods in Corpus Linguistics. In: Lüdeling, Anke/Kytö, Merja (Hg.): *Corpus Linguistics. An International Handbook*. Bd. 2. Berlin: Mouton de Gruyter, S. 1286–1304. Bildhauer, Felix (2011): Mehrfache Vorfeldbesetzung und Informationsstruktur: Eine Bestandsaufnahme. In: *Deutsche Sprache* 39, S. 362–379. (https://bit.ly/2YbmzoC)

Bohnet, Bernd (2010): Top Accuracy and Fast Dependency Parsing is not a Contradiction. In: *The 23rd International Conference on Computational Linguistics* (*COLING 2010*). Peking. (http://aclweb.org/anthology/C10-1011)

Breyer, Yvonne (2009): Learning and teaching with corpora: reflections by student teachers. In: *Computer-Assisted Language Learning* 22 (2), S. 153–172.

Cheung, Jackie C. K./Penn, Gerald (2009): Topological Field Parsing of German. In: *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics* (*ACL*). Singapur, S. 64–72. (http://www.aclweb.org/anthology/P09-1008)

Clark, Alexander/Fox, Chris/Lappin, Shalom (Hg., 2010): *The Handbook of Computational Linguistics and Natural Language Processing*. Oxford: Wiley-Blackwell.

Coniglio, Marco/Schlachter, Eva (2015): The properties of the Middle High German ›Nachfeld‹: Syntax, information structure, and linkage in discourse. In: Gippert, Jost/Gehrke, Ralf (Hg.): *Historical Corpora. Challenges and Perspectives*. Tübingen: Narr, S. 125–136.

Crysmann, Berthold/Hansen-Schirra, Silvia/Smith, George/Ziegler-Eisele, Dorothea (2005): *TIGER-Morphologie-Annotationsschema*. Technischer Bericht. Uni-

versität Saarbrücken, Universität Potsdam. (http://doeplayer.org/33988647-Tiger-morphologie-annotationsschema.html)
Deppermann, Arnulf/Schmidt, Thomas (2014): Gesprächsdatenbanken als methodisches Instrument der Interaktionalen Linguistik – Eine exemplarische Untersuchung auf Basis des Korpus FOLK in der Datenbank für Gesprochenes Deutsch (DGD2). In: Domke, Christine/Gansel, Christa (Hg.): *Korpora in der Linguistik – Perspektiven und Positionen zu Daten und Datenerhebung*. Göttingen: Vandenhoeck & Ruprecht, S. 4–17.
Dietrich, Rainer/Gerwien, Johannes (2017): *Psycholinguistik. Eine Einführung*. 3. Aufl. Stuttgart: Metzler.
Dipper, Stefanie/Faulstich, Lukas/Leser, Ulf/Lüdeling, Anke (2004): Challenges in Modelling a Richly Annotated Diachronic corpus of German. In: *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*. Lissabon, S. 21–29. (https://bit.ly/2FqgpcS)
Donhauser, Karin (2015): Das Referenzkorpus Altdeutsch. In: Gippert, Jost/Gehrke, Ralf (Hg.): *Historical Corpora. Challenges and Perspectives*. Tübingen: Narr, S. 35–49.
Doolittle, Seanna (2008): *Entwicklung und Evaluierung eines auf dem Stellungsfeldermodell basierenden syntaktischen Annotationsverfahrens für Lernerkorpora innerhalb einer Mehrebenen-Architektur mit Schwerpunkt auf schriftlichen Texten fortgeschrittener Deutschlerner*. Magisterarbeit. Humboldt-Universität zu Berlin. (https://edoc.hu-berlin.de/bitstream/handle/18452/14786/doolittle.pdf)
Drach, Erich (1937): *Grundgedanken der Deutschen Satzlehre*. Frankfurt a. M.: Diesterweg.
Eckart de Castilho, Richard/Mújdricza-Maydt, Eva/Yimam, Seid M./Hartmann, Silvana/Gurevych, Iryna/Frank, Anette/Biemann, Chris (2016): A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In: *Proceedings of the LT4DH workshop at COLING 2016*, S. 76–84. (http://www.aclweb.org/anthology/W16-4011)
Evert, Stefan and Hardie, Andrew (2011): Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In: *Proceedings of the Corpus Linguistics 2011 conference*. University of Birmingham. (http://www.stefan-evert.de/PUB/EvertHardie2011.pdf)
Fandrych, Christian/Meißner, Cordula/Wallner, Franziska (2017): *Gesprochene Wissenschaftssprache – digital. Verfahren zur Annotation und Analyse mündlicher Korpora*. Tübingen: Stauffenburg Verlag.
Faruqui, Manaal/Padó, Sebastian (2010): Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In: *Proceedings of KONVENS 2010*. Saarbrücken. (https://bit.ly/2Fitc00)
Foth, Kilian (2006): *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*. Technischer Bericht. Universität Hamburg. (https://bit.ly/2FpUvq9)
Gippert, Jost/Gehrke, Ralf (Hg., 2015): *Historical Corpora. Challenges and Perspectives*. Tübingen: Narr.
Granger, Sylviane (2008): Learner corpora. In: Lüdeling, Anke/Kytö, Merja (Hg.): *Corpus Linguistics. An International Handbook*. Bd. 1. Berlin: de Gruyter, S. 259–275.
Granger, Sylviane/Dagneaux, Estelle/Meunier, Fanny/Paquot, Magali (2009): *International Corpus of Learner English v2*. Louvain-la-Neuve: Presses universitaires de Louvain.
Gries, Stefan Th. (2013): *Statistics for linguistics with R*. 2. Aufl. Berlin/Boston: Mouton de Gruyter.
Gries, Stefan Th. (2008): *Statistik für Sprachwissenschaftler*. Göttingen: Vandenhoeck & Ruprecht.
Heylen, Kris/Speelman, Dirk (2003): A corpus-based analysis of word order variation: The order of verb arguments in the German middle field. In: *Procee-*

dings of the Corpus Linguistics 2003. Lancaster, S. 320–329. (https://bit.ly/2Fq4rQv)

Höhle, N. Tilman (1986): Der Begriff ›Mittelfeld‹: Anmerkungen über die Theorie der topologischen Felder. In: Müller, Stefan/Reis, Marga/Richter, Frank (Hg., 2018): *Beiträge zur deutschen Grammatik*. Gesammelte Schriften von Tilman N. Höhle. Berlin: Language Science Press, S. 279–294. (http://langsci-press.org/catalog/view/149/596/998-2)

Imo, Wolfgang (2008): Individuelle Konstrukte oder Vorboten einer neuen Konstruktion? Stellungsvarianten der Modalpartikel halt im Vor- und Nachfeld. In: Stefanowitsch, Anatol/Fischer, Kerstin (Hg.): *Konstruktionsgrammatik II. Von der Konstruktion zur Grammatik*. Tübingen: Stauffenburg, S. 135–156.

Imo, Wofgang/Lanwer, Jens Philipp (2019): *Interaktionale Linguistik. Eine Einführung*. Stuttgart: J. B. Metzler.

Imo, Wolfgang/Weidner, Beate (2018): Mündliche Korpora im DaF- und DaZ-Unterricht. In: Kupietz, Marc/Schmidt, Thomas (Hg.): *Korpuslinguistik*. Berlin/Boston: de Gruyter, S. 231–251.

Jurish, Bryan (2012): *Finite-state Canonicalization Techniques for Historical German*. Dissertation. Universität Potsdam. (https://bit.ly/2TRwNw4)

Jurish, Bryan/Würzner, Kay-Michael (2013): Word and Sentence Tokenization with Hidden Markov Models. In: *Journal for Language Technology and Computational Linguistics* 28(2), S. 61–83. (https://~~kaskade.dwds.de/~moocow/mirror/pubs/jw2013tokenization_draft.pdf~~)

Krause, Thomas/Zeldes, Amir (2016): ANNIS3: A new architecture for generic corpus query and visualization. In: *Digital Scholarship in the Humanities* 2016 (31). (https://bit.ly/2JumGIR)

Lemnitzer, Lothar/Zinsmeister, Heike (2015): *Korpuslinguistik. Eine Einführung*. 3. Aufl. Tübingen: Gunter Narr Verlag.

Levshina, Natalia (2015): *How to do Linguistics with R: Data Exploration and Statistical Analysis*. Amsterdam/Philadelphia: John Benjamins.

Lezius, Wolfgang (2000): Morphy – German Morphology, Part-of-Speech Tagging and Applications. In: *Proceedings of the 9th EURALEX International Congress*. Stuttgart, S. 619–623.

Lüdeling, Anke (2007): Das Zusammenspiel von qualitativen und quantitativen Methoden in der Korpuslinguistik. In: Kallmeyer, Werner/Zifonun, Gisela (Hg.): *Sprachkorpora. Datenmengen und Erkenntnisfortschritt*. Berlin/New York: de Gruyter, S. 28–48.

Lüdeling, Anke/Doolittle, Seanna/Hirschmann, Hagen/Schmidt, Karin/Walter, Maik (2008): Das Lernerkorpus Falko. In: *Deutsch als Fremdsprache* 45 (2), S. 67–73.

Lüdeling, Anke/Kytö, Merja (Hg., 2009): *Corpus Linguistics. An International Handbook*. Bd. 2. Berlin: de Gruyter.

Lüdeling, Anke/Kytö, Merja (Hg., 2008): *Corpus Linguistics. An International Handbook*. Bd. 1. Berlin: de Gruyter.

Mann, William C./Thompson, Sandra A. (1988): Rhetorical Structure Theory: Toward a functional theory of text organization. In: *Text* 8 (3), 243–281. (http://www.cis.upenn.edu/~nenkova/Courses/cis700-2/rst.pdf)

Mukherjee, Joybrato (2009): *Anglistische Korpuslinguistik. Eine Einführung*. Berlin: Erich Schmidt Verlag.

Müller, Stefan (2013): *Datensammlung zur scheinbar mehrfachen Vorfeldbesetzung*. Materialsammlung. Freie Universität Berlin. (https://hpsg.hu-berlin.de/~stefan/PS/mehr-vf-daten.pdf)

Müller, Stefan (2005): Zur Analyse der scheinbar mehrfachen Vorfeldbesetzung. In: *Linguistische Berichte* 203, S. 297–330. (https://hpsg.hu-berlin.de/~stefan/PS/mehr-vf-lb.pdf)

Müller, Stefan (2003): Mehrfache Vorfeldbesetzung. In: *Deutsche Sprache* 31(1), S. 29–62. (https://hpsg.hu-berlin.de/~stefan/PS/vorfeld-ds2003.pdf)

Müller, Stefan/Bildhauer, Felix/Cook, Philippa (2012): Beschränkungen für die scheinbar mehrfache Vorfeldbesetzung im Deutschen. In: Cortès, Colette (Hg.): *Satzeröffnung. Formen, Funktionen, Strategien*. Tübingen: Stauffenburg, S. 113–128. (https://hpsg.hu-berlin.de/~stefan/PS/mehr-vf-beschraenkungen.pdf)

Müller, Christoph/Strube, Michael (2006): Multilevel annotation of linguistic data with MMAX2. In: Braun, Sabine/Kohn, Kurt/Mukherjee, Joybrato (Hg.): *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Frankfurt a. M.: Peter Lang, S. 197–214. (https://bit.ly/2TWgKwh)

Nivre, Joakim/Hall, Johan/Nilsson, Jens (2006): MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (LREC 2006), S. 2216–2219. (https://bit.ly/2Jugxw2)

O'Donnell, Michael (1997): RST-Tool: An RST Analysis Tool. In: *Proceedings of the 6th European Workshop on Natural Language Generation*. Duisburg.

Odebrecht, Carolin/Belz, Malte/Zeldes, Amir/Lüdeling, Anke/Krause, Thomas (2016): RIDGES Herbology – Designing a Diachronic Multi-Layer Corpus. In: *Language Resources and Evaluation 51* (3), S. 695–725.

Perkuhn, Rainer/Keibel, Holger/Kupietz, Marc (2012): *Korpuslinguistik*. Paderborn: Fink.

Peters, Robert (2017): Das Referenzkorpus Mittelniederdeutsch/Niederrheinisch (1200–1650). In: *Niederdeutsches Jahrbuch. Jahrbuch des Vereins für niederdeutsche Sprachforschung* 140, S. 35–42.

Petran, Florian/Bollmann, Marcel/Dipper, Stefanie/Klein, Thomas (2016): ReM: A reference corpus of Middle High German – corpus compilation, annotation, and access. In: *Journal for Language Technology and Computational Linguistics* 31 (2), S. 1–15.

Reznicek, Marc/Lüdeling, Anke/Krummes, Cedric/Schwantuschke, Franziska/Walter, Maik/Schmidt, Karin/Hirschmann, Hagen/Andreas, Torsten (2012): *Das Falko-Handbuch. Korpusaufbau und Annotationen. Version 2.01*. Technischer Bericht. Humboldt-Universität zu Berlin. (https://hu.berlin/falko-handbuch)

Sauer, Simon/Lüdeling, Anke (2016): Flexible Multi-Layer Spoken Dialogue Corpora. In: *International Journal of Corpus Linguistics* 21 (3), S. 419–438. (https://hu.berlin/bematac-paper)

Schäfer, Roland (2016a): *Einführung in die grammatische Beschreibung des Deutschen*. Zweite, überarb. Aufl. Berlin: Language Science Press. (http://langsci-press.org/catalog/view/46/25/232-1)

Schäfer, Roland (2016b): CommonCOW: massively huge web corpora from CommonCrawl data and a method to distribute them freely under restrictive EU copyright laws. In: *Proceedings of LREC 2016*, S. 4500–4504. (https://bit.ly/2Jt3eMk)

Schalowski, Sören (2015): *Wortstellungsvariation aus informationsstruktureller Perspektive: Eine Untersuchung der linken Satzperipherie im gesprochenen Deutsch*. Potsdam: Universitätsverlag Potsdam. (https://bit.ly/2uh3tQq)

Scherer, Carmen (2006): *Korpuslinguistik*. Heidelberg: Universitätsverlag Winter.

Schiller, Anne/Teufel, Simone/Stöckert, Christine/Thielen, Christine (1999): *Guidelines für das Tagging deutscher Textcorpora mit STTS*. Technischer Bericht. Institut für maschinelle Sprachverarbeitung, Stuttgart. (http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf)

Schmid, Helmut (1995): Improvements in Part-of-Speech Tagging with an Application to German. In: *Proceedings of the ACL SIGDAT-Workshop*. Dublin.

Schmid, Helmut (1994): Probabilistic part-of-speech tagging using Decision Trees. In: *Proceedings of the International Conference on New Methods in Language Processing*. (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf)

Schmid, Helmut/Fitschen, Arne/Heid, Ulrich (2004): SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In: *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*. Lissabon, S. 1263–1266. (http://www.cis.uni-muenchen.de/~schmid/papers/smor.pdf)

Schmidt, Thomas (2012): EXMARaLDA and the FOLK tools – two toolsets for transcribing and annotating spoken language. In: *Proceedings of LREC 2010*. (http://www.lrec-conf.org/proceedings/lrec2012/pdf/529_Paper.pdf)

Schmidt, Thomas/Schütte, Wilfried/Winterscheidt, Jenny (2015): *cGAT. Konventionen für das computergestützte Transkribieren in Anlehnung an das Gesprächsanalytische Transkriptionssystem 2 (GAT2)*. Technischer Bericht. Mannheim, Institut für Deutsche Sprache (IDS). (http://agd.ids-mannheim.de/download/cgat_handbuch_version_1_0.pdf)

Schmidt, Thomas/Wörner, Kai (2014): EXMARaLDA. In: *Handbook on Corpus Phonology*. Oxford: Oxford University Press, S. 402–419.

Selting, Margret/Auer, Peter/Barth-Weingarten, Dagmar/Bergmann, Jörg R./Bergmann, Pia/Birkner, Karin/Couper-Kuhlen, Elizabeth/Deppermann, Arnulf/Gilles, Peter/Günthner, Susanne/Hartung, Martin/Kern, Friederike/Mertzlufft, Christine/Meyer, ChristianUniBi/Morek, Miriam/Oberzaucher, Frank/Peters, Jörg/Quasthoff, Uta/Schütte, Wilfried/Stukenbrock, Anja/Uhmann, Susanne (2009): Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). In: *Gesprächsforschung – Online-Zeitschrift zur verbalen Interaktion* 10, S. 353–402. (http://www.gespraechsforschung-ozs.de/heft2009/px-gat2.pdf)

Selting, Margret/Auer, Peter/Barden, Birgit/Bergmann, Jörg R./Couper-Kuhlen, Elizabeth/Günthner, Susanne/Meier, Christoph/Quasthoff, Uta M./Schlobinski, Peter/Uhmann, Susanne (1998): Gesprächsanalytisches Transkriptionssystem (GAT). In: *Linguistische Berichte* 173. S. 91–122. (http://www.mediensprache.net/de/medienanalyse/transcription/gat/gat.pdf)

Sennrich, Rico/Volk, Martin/Schneider, Gerold (2013): Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing*, S. 601–609. (http://www.aclweb.org/anthology/R13-1079)

Stede, Manfred (Hg., 2016): *Handbuch Textannotation. Potsdamer Kommentarkorpus 2.0*. Potsdam: Universitätsverlag Potsdam. (https://bit.ly/2W9hnje)

Stede, Manfred (2007): *Korpusgestützte Textanalyse. Grundzüge der Ebenen-orientierten Textlinguistik*. Tübingen: Narr.

Stefanowitsch, Anatol/Gries, Stefan Th. (2003): Collostructions: investigating the interaction between words and constructions. In: *International Journal of Corpus Linguistics* 8 (2), S. 209–243. (http://www.stgries.info/research/2003_AS-STG_Collostructions_IJCL.pdf)

Steinbach, Markus/Albert, Ruth/Girnth, Heiko/Hohenberger, Annette/Kümmerling-Meibauer, Bettina/Meibauer, Jörg/Rothweiler, Monika/Schwarz-Friesel, Monika (2007): *Schnittstellen der germanistischen Linguistik*. Stuttgart/Weimar: J. B. Metzler.

Szmrecsanyi, Benedikt/Wälchli, Bernhard (Hg., 2014): *Aggregating dialectology, typology, and register analysis. Linguistic variation in text and speech*. Berlin: de Gruyter.

Telljohann, Heike/Hinrichs, Erhard/Kübler, Sandra/Zinsmeister, Heike (2017): *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Technischer Bericht. Universität Tübingen. (http://www.sfs.uni-tuebingen.de/fileadmin/static/ascl/resources/tuebadz-stylebook-1707.pdf)

Tesnière, Lucien (1980): *Grundzüge der strukturalen Syntax. Herausgegeben und übersetzt von Ulrich Engel*. Stuttgart: Klett.

Tuggener, Don/Klenner, Manfred (2012): A Hybrid Entity-Mention Pronoun Resolution Model for German Using Markov Logic Networks. In: *KONVENS 2014*.

Hildesheim, S. 21–29. (http://www.zora.uzh.ch/id/eprint/99594/1/konvens2014_german_pronouns_res_MLNs.pdf)

Witten, Ian H./Frank, Eibe/Hall, Mark A. (2011): *Data Mining. Practical Machine Learning Tools and Techniques*. 3. Aufl. Amsterdam: Morgan Kaufmann.

Zeldes, Amir (2016): rstWeb – A Browser-based Annotation Interface for Rhetorical Structure Theory and Discourse Relations. In: *Proceedings of NAACL-HLT 2016 System Demonstrations*. San Diego. (http://www.aclweb.org/anthology/N16–3001)

Zeng, Anne-Christin (2016): *Die Polyfunktionalität des Ausdrucks wohl. Eine korpuslinguistische Untersuchung der Bedeutung und Verwendung der Modalpartikel* wohl *im Vorfeld in einem Korpus aus Parlamentsreden*. Bachelorarbeit. Humboldt-Universität zu Berlin. (https://edoc.hu-berlin.de/bitstream/handle/18452/14921/zeng.pdf)

Zipser, Florian/Romary, Laurent (2010): A model oriented approach to the mapping of annotation formats using standards. In: *Proceedings of the Workshop on Language Resource and Language Technology Standards* (LREC 2010). (https://hal.inria.fr/inria-00527799/document)

5.3 | Sachregister

A
Abfragesprache *siehe* Anfragesprache
Abfragesyntax *siehe* Anfragesyntax
Accuracy *siehe* Akkuratheit
Adjazenz 131, 213–214, 217
adjudizieren 99
Ähnlichkeitsmaß 215
Akkuratheit 96, 101–102
ALC-Korpus 6
Alignement *siehe* Alignierung
alignieren *siehe* Alignierung
Alignierung 46, 85, 137
Ambiguität 41
Analyserichtlinien 96
Anfragesprache 15, 105, 108–109, 111, 117, 130, 140, 153, 166
Anfragesyntax 105, 108, 112, 131
ANNIS-Suchinterface *siehe* ANNIS (Suchwerkzeug)
ANNIS (Suchwerkzeug) 11–12, 55, 68, 86, 105, 107–108, 114, 149–150, 152, 179, 182, 187, 195–196, 203, 219–220
Annotate (Annotationswerkzeug) 67
Annotation 2–4, 7, 19, 22–24, 27, 29, 31, 34, 42, 75
– Evaluation von Annotationen 100
– Gütekriterien 101
– Inline-Annotation 36, 42
– Mehrebenenannotation 45
– textlinguistische 68, 70
– tokenbasierte 34
– und Interpretation 2
– von Akzentstrukturen 86
– von Anaphern 68
– von Artikulationslänge 87
– von Dependenzstrukturen 63–64
– von Dialogizität 84
– von Diskursstruktur 70, 147
– von Eigennamen 49
– von Fillern 86
– von Flexionsmorphologie 44
– von Ko- und Diskursreferenz 68–69
– von Lautabfolgen 82
– von Lemmata 34, 64
– von Pausen 86
– von Prosodie 86
– von RST-Strukturen 71
– von Satzspannen 54
– von Semantik 72
– von Syntax 52, 60
– von Tonhöhenverläufen 86
– von topologischen Feldern 56
– von Turninteraktionen 84
– von Wortarten 34, 64
– vs. Normalisierung 29
– vs. Transkription 75
– wortbildungsmorphologische 47
Annotationsebene 78, 105, 110, 120, 129
Annotationsformat 42, 66–67
Annotationsguidelines *siehe* Annotationsrichtlinien
Annotationsrichtlinien 22, 56
– für die Transkription 76
– für Diskursstruktur 71
– für Gesprächskorpora 83
– für Syntax 51, 66, 68
– für topologische Felder 56–57
– für Wortarten 36
– textlinguistische 68
Annotationsschema 57, 66
Annotationsschritt 23, 35
Annotationsspanne 135
Annotationstyp 22
Annotationswerkzeug 23
– für das Tagging 39
– für die Normalisierung 94
– für Flexionsmorphologie 45
– für syntaktische Annotationen 57
– für Wortbildung 48
AntConc (Such- und Auswertungswerkzeug) 105, 107, 184
Anvil (Transkriptionswerkzeug) 78
AQL (ANNIS Query Language) 108, 131
Arborator (Annotationswerkzeug) 63
Architektur 45, 48, 119
arithmetisches Mittel 200–201
Arity 144
Assoziation 215, 217
Assoziationsmaß 215
Assoziationswert 215
Audiosignal 76
Ausgewogenheit 17
Authentizität 3, 14

B
Balkendiagramm 190
Basistranskript 86, 91
Baum 60
– Dependenzstrukturbaum 61
– Kante 63
– Knoten 61
– Konstituentenstrukturbaum 61
– Phrasenstrukturbaum 61

Baumbank 60
Baumstruktur 61
Belegressource 14
Belegsammlung 15
BeMaTaC-Korpus 12, 86, 95, 187, 195, 203
Berkeley-Parser 57, 67
Beschreibungsebene 23
Blatt (Blattknoten) 60
Brat (Annotationswerkzeug) 63

C
CAB (Normalisierungswerkzeug) 94
CALL (Computer-Assisted Language Learning) 15
case (in)sensitive 126
cGAT 83
Chi-Quadrat-Test 204
Chi-Quadrat-Wert 204
Chunk 52, 60
Chunking 52, 54, 59
CIA (Contrastive Interlanguage Analysis) 12
CoNLL-Format 63–64, 70
Corpus Workbench 54
CorZu (Annotationswerkzeug) 70
COSMAS (Suchwerkzeug) 17, 21, 105, 162
CQP (Suchwerkzeug) 108, 114
CSV-Kodierung 193
CWB *siehe* Corpus Workbench

D
Datenarchitektur 45, 48
Datenformat 42
Datenkonversion 178
Datenstreuung 200
DDD (Verbundsinitiative Deutsch diachron digital) 17
DDL (Data-Driven Learning) 15
Depency Viewer (Visualisierungswerkzeug) 64
Dependenzannotation 64, 70
Dependenzstruktur 61–62, 64
Dependenzstrukturbaum 61
Dependenzstrukturbaumbank 144, 146
DeReKo (Deutsches Referenzkorpus) 17, 21
Deutsch als Fremdsprache *siehe* Lernerkorpus
DG Annotator (Werkzeug für Dependenzannotationen) 64
DGD (Datenbank gesprochenes Deutsch) 93, 105, 168
Diagramm (Typen) 190
Dice-Koeffizient (Ähnlichkeitsmaß) 215
Dominanz 61
DTA (Deutsches Textarchiv) 11, 105, 157
DWDS (Digitales Wörterbuch der deutschen Sprache) 13, 105, 157, 192

E
Edge *siehe* Kante
EDU (Elementary Discourse Unit) 70
Eigennamenerkennung *siehe* NER (Named Entity Recognition)
ELAN (Transkriptions- und Annotationswerkzeug) 24, 78, 85
empirische Methode 1
Error Analysis 12
Evaluation
– von Annotationen 96
– von Korpussuchen 172
EXMARaLDA (Transkriptions- und Annotationswerkzeug) 79, 135
Export (bei der Korpussuche) 155
Exporter (bei der Korpussuche) 152–153

F
Falko-Korpus 12, 21, 57, 150
false positive (falscher Treffer) 101
Fehleranalyse 12
Flexion (und Normalisierung) 29
F-measure *siehe* F-score
FOLKER (Transkriptionswerkzeug) 85, 94
FOLK (Korpus) 6, 17, 82–83, 93, 168
Frequenz *siehe* Häufigkeit
Frequenzauswertung 182
Frequenzliste 182
F-score 100–102, 175
Fürstinnenkorrespondenzkorpus 55

G
GAT 2 76
GAT (Gesprächsanalytisches Transkriptionssystem) 76, 82, 91
GAT-Transkription 78
Gesprächskorpus 74, 83, 187
gesprochene Sprache 22, 30, 74
GeWiss-Korpus 12, 82
Gleichheitstest 204
Goldstandard 97, 100
grafische Nutzeroberfläche 109
Grammatikalität 8, 14, 57
Grammatikforschung 7
Ground Truth 100

Guidelines *siehe* Annotationsrichtlinien
GUI (graphical user interface) *siehe* grafische Nutzeroberfläche

H
Häufigkeit 6, 150, 153, 177, 182, 186–187, 190, 193, 204, 208, 215
– absolute Häufigkeit 186–188, 193–195, 203
– gemittelte Häufigkeit 193
– normalisierte Häufigkeit 188, 203
historische Korpora 10, 157
Homogenisierung 29
Homogenität 17
Homogenitätstest 204

I
IAA *siehe* Inter-Annotator-Agreement
ICLE (International Corpus of Learner English) 12
Inline-Annotation 36, 42
Inline-XML 24
Inter-Annotator-Agreement 97, 101
Interface *siehe* Suchinterface
IPA (Internationales Phonetisches Alphabet 76

K
Kante 60
Kantenbezeichnung 63
Key word in context *siehe* KWIC
Klammerformat 67
Knoten 60
– Mutterknoten 61
– Schwesterknoten 61
– terminaler Knoten 60
– Tochterknoten 61
– unärer Knoten 60
– Wurzelknoten 60, 64
KoGra-R (Auswertungswerkzeug) 193
Kolligation 213
Kollokation 213, 215, 217
Kollostruktion 214
Konfidenzintervall 198–199
Konstituente 61, 66
Konstituentenstruktur 64, 67–68
Konstituentenstrukturbaum 61
kontrastive Analyse 186
Konversion (von Korpusdaten) *siehe* Datenkonversion
Konverter 68
Kookkurrenz 131, 213–214
KorAP (Suchwerkzeug) 105, 162
Koreferenzannotation 69–70, 148
Korpusarchitektur 45
Korpusdaten (Einordnung des Datentyps) 4
Korpus (Definition) 2
Korpusgröße 186
– bei der Berechnung von Assoziation 215
– bei der Normalisierung 186–187
Korpuslinguistik (Definition) 1
Korpussuche *siehe* Suche
Korrelation 192, 218–219
Kreisdiagramm 191
KWIC (Key Word in Context) 107, 165, 178

L
Längsschnittstudie 185
Layer *siehe* Annotationsebene
Lemma 120
Lemmasuche 120
Lemmatisierung 37–39, 120
Lernerkorpus 11, 57, 108, 150, 203
Lexikographie 13, 157
Lexikon (bei Taggern) 41
Lexik (und Normalisierung) 29
Linguee (Online-Übersetzungsdienst) 13, 15
literarische Umschrift 82

M
MaltParser 63
Märchenkorpus 151
Mate-Tools (Annotationswerkzeuge) 63
Maximalphrase 58
Median 200–201
Mehrebenenannotation 45
Mehrebenenarchitektur 119
Mengensuche 133
Merging 31
MERLIN (Plattform zur Analyse von Lernersprache) 108
Metadaten 2–4, 102–103, 149
– vs. Annotation 103
Minimaltranskript 82
Mittelwert 194, 196–197, 199, 201–202, 205–206
Mittelwertabweichung 197
MMAX2 (Annotationswerkzeug) 69
Morphy (Annotationswerkzeug) 48
Multilayer-Architektur *siehe* Mehrebenenarchitektur
Mündlichkeit 10, 21, 30, 170
Mustersuche 117, 124–128, 133
Mutterknoten 61
Mutual Information (Assoziationsmaß) 215–217

N
Negation (in der Korpussuche) 133
Negationsoperator 133
NER (Named Entity Recognition) 50
N-Gramm 131
NLP (Natural Language Processing) 15
Node *siehe* Knoten
Normalisierung
- bei der Korpusauswertung 187–188, 193–195, 203–204
- bei der Korpuserstellung 29–30, 73, 75–76, 78, 81, 93–94, 169
- vs. Annotation 29
Normalisierungsgröße 187, 203, 205
Normalisierungsprogramm 94
Normalisierungswert 188, 193, 195–196, 204
NoSketch Engine (Suchwerkzeug) 108, 110, 114
Nutzeroberfläche 106

O
OCTRA (Transkriptionswerkzeug) 78
Online-Suchinterface 156
Optionalität (in der Korpussuche) 127
Organisationsname 50
Orthographie (und Normalisierung) 29
OrthoNormal (Normalisierungswerkzeug) 94
Ortsname 50

P
Parallelkorpus 13
Parlamentsredenkorpus 21, 113, 149, 154, 210
Parser 63, 67
Parsing 52, 58, 63
Partitureditor 23
ParZu (Annotationswerkzeug) 63
PCC (Potsdam Commentary Corpus) 68, 148
PENN-Klammerformat 67
PENN-Treebank 67
Pepper (Konversionswerkzeug) 68, 70
Personenname 50
Phrase 66
Phrasenstrukturbaum 61
Phrasenstrukturbaumbank 138, 140, 142, 144
Plaintextformat 42
Platzhalter (Suchoperator) 117
Pointing Relation 69, 147
pos-Tagging 35
PRAAT (Aufnahme- und Analysewerkzeug) 78
Präzedenz 131
Precision 100, 172
Primärdaten 2–3, 19–20, 22–23, 31, 79, 102–103, 115
- Kritik am Konzept 20
prop.test 205
Punktdiagramm 192
p-Wert 202, 204

Q
qualitative Forschung 6
quantitative Forschung 6
Querschnittstudie 185
Query Language *siehe* Anfragesprache

R
Recall 100, 174
Reference relation 147
Referenzkorpus 17
Referenzkorpus Alt-, Mittelhoch-, Mittelnieder-, Frühneuhochdeutsch 10, 17
Referenzkorpus Mittelhochdeutsch 108, 212
regulärer Ausdruck 117, 124–128, 133–134
Repräsentativität 16
RFTagger (Annotationswerkzeug) 45
Richtlinien *siehe* Annotationsrichtlinien
RIDGES-Korpus 11
Root *siehe* Wurzel (Wurzelknoten)
R-Script 206
RST-Annotation 70–71
RSTTool (Annotationswerkzeug) 72
RStudio (Auswertungs-Nutzeroberfläche) 200, 204–206
RstWeb (Annotationswerkzeug) 72

S
Satzsegmentierung 54
Satzspanne 54, 135
Säulendiagramm 190
Schnittstelle *siehe* Suchinterface
Schwesterknoten 61
Segmentierung 84, 91
sekundäre Kante 68
shallow parsing 58
Signifikanz 202–203
- Signifikanzniveau 206
- Signifikanzwert 206
SMOR (Annotationswerkzeug) 46, 48
Spanne 60
Spannenannotation 52, 136
Speicherformat 42, 64
Spracherwerbsforschung 11
Standardabweichung 197

Standardkorpus 156
Standoff-XML 24
Stanford-Parser 67
Statistik (Typen) 181
statistische Testverfahren 206
Stichprobe 205
Streudiagramm 192
Streuung 200
String *siehe* Zeichenkette
STTS-Tagset 36, 40, 51, 63, 73, 187
Studientyp (A, B und C) 185
Subkorpus 185, 197
Subtoken 31
Suche 119
– Evaluation von Korpussuchen 172
– mit bestimmten Abständen 131
– mit Metadaten 149
– mit Negation 133
– nach Abfolgen 131
– nach Anaphern 147
– nach Annotationsspannen 136
– nach Flexionskategorien 122
– nach Kantenannotationen 142, 146
– nach Konstituenten 138
– nach Koreferenz 147
– nach Lemmata 120
– nach Mustern 124
– nach optionalen Elementen 127
– nach Phrasen 138
– nach syntaktischen Funktionen 142, 146
– nach syntaktischen Relationen 140, 144
– nach topologischen Feldern 135
– nach Wortarten 121
– nach Wortformen 115
– nach Zeichenmengen 125, 133–134
Suchinterface 105–106, 109
Suchmuster 133
Suchoperator 117
Suchprogramm 106
Suchwerkzeug 106, 114
Syntaxbaum 60

T
tabellarische Auswertung 190
Tag (Annotationskürzel) 37
TagAnt (Annotationswerkzeug) 39
Tagger 39–40, 94, 101
Tagging 34, 101
Tagset 96
TCF (Datenformat) 73
terminaler Knoten 61
Tier (Annotationsebene in EXMARaLDA) 80
TIGER-Annotationsformat 66
TIGER-Annotationsschema 66
TIGER-Korpus 9, 50, 67, 69, 111, 123, 139, 142, 211, 219
TIGERSearch (Suchwerkzeug) 108, 111, 115
TIGER-XML 67
Tochterknoten 61
Token 31
Tokenannotation 34
Tokenarity 144
Tokendefinition (variable) 32
Tokenisierer (Werkzeug zur Tokenisierung) 33, 54
Tokenisierung 31–32, 47, 84, 93, 95
Tokenisierungsseparator 32
Tokenisierung vs. Annotation 31
Tokenizer *siehe* Tokenisierer
topologisches Feld 56, 135
Topoparser 57
Transcriber (Transkriptionswerkzeug) 78
transkribieren *siehe* Transkription
Transkription 22, 75–76, 78, 84, 86, 92
– Transkription vs. Annotation 75
– Transktiption und Normalisierung 76
Transkriptionsrichtlinien 76
Transkriptionssegment 85
Transkriptionswerkzeug 78
Treebank 60
TreeTagger (Annotationswerkzeug) 33, 39, 42
Trefferexport 152–153, 155
t-Test 206
TüBa-D/Z-Annotationsformat 66
TüBa-D/Z-Korpus 9, 56–57, 66–67, 147, 208
TüNDRA (Suchwerkzeug) 108, 112, 115

U
UAM Corpus Tool (Annotationswerkzeug) 72
Übersetzungskorpus 13
unärer Knoten 60

V
Valenzmodell 62
Variable 119, 171, 191, 208, 219
Variablen-Wert-Paar 45, 119
Variante 208–209
Varianz 192, 194, 199, 202
Varietät 9, 208
Vergleichsprogramm 100
Verlaufsdiagramm 191

W
WASTE-Tokenizer 54
WebAnno (Annotationswerkzeug) 63, 69
WebLicht (Annotationsplattform) 27, 33, 39, 46, 51, 54, 57, 67–68, 73, 94
Wert 119
What's Wrong With My NLP? (Visualisierungswerkzeug) 64
Wildcard 117
Wortartenanalyse 35
Wortartensuche 121
Wortartentagging 39
Wörterbucheintrag 157
Wortstatistik 192
Wurzel (Wurzelknoten) 60–61

X
XML-Format 67, 70
XML-Kodierung 43

Z
Zählung 187
Zeichenkette 133–134
Zweistichprobentest 204