Reinhard Köhler

Korpuslinguistik – zu wissenschaftstheoretischen Grundlagen und methodologischen Perspektiven

1 Einführung

Im Zusammenhang mit den folgenden Überlegungen steht der Terminus *Korpuslinguistik* für die Gesamtheit aller Tätigkeiten, die darauf gerichtet sind, (1) umfangreiches authentisches Sprach- oder Textmaterial (gesprochen oder geschrieben) zu sammeln, zusammen zu stellen, aufzubereiten, mit Informationen zu annotieren, zu verwalten und zu warten sowie verfügbar zu machen, (2) solches Material für wissenschaftliche oder technische Zwecke oder andere Anwendungen systematisch auszuwerten.

Das oft konstatierte, wachsende Interesse an Korpus-basierten Ansätzen hat verschiedene Gründe. Zunächst waren Vorbedingungen für die zunehmende Erstellung bzw. Verwendung von großen maschinenoperablen Textkorpora Fortschritte in der Hard- und Softwaretechnik sowie leistungsstarke Verfahren der Sprachtechnologie. Die heutige Hardware-, Software- und Netzwerktechnik erleichtern Digitalisierung, elektronische Produktion, Speicherung und Verbreitung von großen Textmengen und sichern somit die Verfügbarkeit von Sprachkorpora. Sprachtechische Verfahren ermöglichen die Indizierung, (teil-)automatische linguistische Annotation sowie effektive Zugriffs- und Abfragesysteme.

Mit der Verfügbarkeit großer und größter Materialsammlungen wurde die früher übliche intellektuelle Inspektion von Texten nach und nach durch die Verwendung statistischer Verfahren abgelöst. Der Durchbruch für die quantitativ-empirischen Ansätze in der maschinellen Sprachverarbeitung kam u. a. mit den Erfolgen der Hidden-Markov-Modelle in Systemen zur Verarbeitung gesprochener Sprache. Doch auch in anderen Bereichen der Sprachtechnik konnten bereits bald viel versprechende Ergebnisse durch den Einsatz statistischer Verfahren erzielt werden. Heute gibt es kaum ein Anwendungsfeld der Computerlinguistik, in dem statistische Methoden nicht – in Kombination mit der oder als Alternative zur diskret-symbolischen Verarbeitung – eine wichtige Rolle spielen.

Wissenschaftstheoretisch betrachtet sind große Mengen von Sprachdaten und ihre statistische Auswertung unverzichtbar für das Überprüfen von Hypothesen, da sprachliche und textuelle Erscheinungen nur in Ausnahmefällen ausreichend mit Hilfe rein formaler Ansätze erfasst werden können. Neben den wissenschaftstheoretischen Einsichten hat dies besonders das praktische Scheitern computerlinguistischer Ansätze, die allein auf formalen Grammatiken u. ä. beruhen, zu genüge gezeigt. Vagheit, Unschärfe, Indeterminiertheit, Variabilität, Dynamik etc. sind Charakteristika der Sprache, die nur durch quantitative Begriffe und Modelle adäquat abgedeckt werden können. Dazu kommt die in vielen Fällen prinzipiell bestehende Unmöglichkeit, den jeweiligen Untersuchungsge-

genstand vollständig zu erfassen – entweder weil er unendlich ist[1] (wie die Menge aller Sätze oder Texte) oder weil sich der Gegenstand während der laufenden Untersuchung verändert (z. B. die Lexik). Gültige Schlussfolgerungen trotz unvollständiger Information mit wählbarer Verlässlichkeit zu ermöglichen, ist gerade die Domäne der Statistik. Dies ist einer der Gründe, aus denen sich die *Quantitative Linguistik* als eigenständige Disziplin herausgebildet hat (s. u.).

Dabei kommt der Analyse realer Sprachdaten eine für die Sprachwissenschaft als empirischer Wissenschaft insofern fundamentale Bedeutung zu, als es prinzipiell keine andere Quelle linguistischer Evidenz gibt als das Sprachverhalten. *Sprache* ist eine Abstraktion. Das System Sprache ist daher auf keine Weise beobachtbar. Auch die mit Sprache und Sprachverarbeitung verbundenen kognitiven Vorgänge sind selbst nicht beobachtbar. Zwar beharren – im Gegensatz zu sprachwissenschaftlichen Disziplinen wie der Typologie und der Universalienforschung, der historischen Linguistik etc. – bestimmte Strömungen der Linguistik auf der Introspektion geschulter Muttersprachler als wichtige (wenn nicht sogar einzige) Möglichkeit, Sprachevidenz zu gewinnen. Allerdings kann dieser Quelle trotz des unbestreitbaren potentiellen heuristischen Werts introspektiv gewonnener Vermutungen auf keinen Fall der Status von empirischen Daten zugebilligt werden. Als Evidenzquellen kommen die folgenden Formen der Beobachtung in Frage:

(a) direkte Beobachtung authentischen (mündlichen oder schriftlichen) Sprachverhaltens;

(b) indirekte Beobachtung durch Analyse von protokolliertem authentischen Sprachverhalten, d. h. Auswertung von schriftlichen Texten oder verschriftlichten oralen Äußerungen;

(c) direkte Beobachtung von manipuliertem/evoziertem Sprachverhalten (z. B. in Form psycholinguistischer Experimente);

(d) indirekte Beobachtung manipulierten/evozierten Sprachverhaltens durch Auswertung von Protokollen bzw. Produkten der Experimente.

Die ersten beiden Möglichkeiten lassen das Studium unverfälschter, natürlichsprachlicher Kommunikation zu, haben aber den Nachteil, dass bestimmte interessierende Phänomene evtl. selten vorkommen und die Datenerhebung ein langes, aufwändiges Vorhaben werden kann. Die beiden letzten lassen umgekehrt die gezielte, schnelle und fast beliebig wiederholbare Beschaffung von Daten zu aktuell interessierenden Fragen zu, bergen aber das Risiko, entscheidende Umstände für das Zustandekommen des jeweiligen Sprachverhaltens so zu verändern, dass Schlussfolgerungen aus den Resultaten der Experimente auf authentisches Sprachverhalten nicht gültig sind. Direkte Beobachtungen, also (a) und (c), haben den Vorteil, dass alle Umstände der Kommunikationssituation mitbeobachtet werden können, während (b) und (d), die indirekten Beobachtungsformen, nur diejenigen Umstände auszuwerten erlauben, die dokumentiert wurden. Andererseits

[1] Rekursive Ansätze ändern daran nichts; denn diese können zwar unendlich viele Ausdrücke beschreiben, aber die Übereinstimmung dieser unendlich vielen hypothetischen Objekte mit den unendlich vielen tatsächlich möglichen kann natürlich nicht überprüft werden.

lässt sich das manifeste Sprachmaterial aus (b) und (d) beliebig oft analysieren. Die Abwägung der sich aus den Zielen einer Untersuchung ergebenden Erfordernisse mit den zur Verfügung stehenden Ressourcen und wissenschaftspraktischen Vor- und Nachteilen führt zu der Entscheidung, welchem Untersuchungstyp jeweils der Vorzug zu geben ist. Neben den eben genannten Kriterien ist ein weiteres von entscheidender Relevanz: das der Datenmenge. So lassen sich auch seltene Ereignisse ausreichend oft beobachten, wenn die Grundmenge der untersuchten Sprachdaten groß genug ist. Vor allem aber ist die Datenmenge ausschlaggebend für die Anwendbarkeit statistischer Methoden (s. o.). Aus diesen Gründen ist heute die Korpuslinguistik als ein spezieller Typ der Beobachtungsformen (b) und (d) verbreitet, der sich durch die Beschaffung und Verwendung großer Mengen manifester Sprachdaten definiert. In allen Fällen, in denen nicht bereits einzelne oder wenige individuelle Beobachtungen für eine Fragestellung ausschlaggebend sein können, stellen Korpusdaten eine Abbildung der sprachlichen Realität dar, an der sich ja alle Hypothesen und Modelle messen und bewähren müssen. Dies gilt für die theoretische linguistische Forschung ebenso wie für die Anwendungen linguistischer Modelle in der maschinellen Sprachverarbeitung, der Sprachdidaktik usw.

2 Theoretische Basis der Korpuslinguistik

Das nicht Theorie-geleitete Arbeiten an Lösungen, die für praktische Ziele der Sprachtechnik erforderlich sind, ist sowohl in der computerlinguistischen Industrie als auch an Universitäten die am weitesten verbreitete korpuslinguistische Aktivität. Diese (legitime) Praxis befindet sich in Entsprechung zu anderen Ingenieurbereichen. Wenn es um technische Applikationen geht, steht im Vordergrund, ob das jeweilige System die erwartete Leistung bringt, und nicht unbedingt, in wie weit die verwendeten Methoden theoretisch gerechtfertigt sind. In diesen Feldern wird nicht Hypothesen-geleitet geforscht; die eingesetzten Verfahren beruhen nicht (unbedingt) auf linguistischer oder kognitiv-psychologischer Fundierung von Sprach(verarbeitungs)modellen, speziell nicht auf wissenschaftstheoretisch reflektierter (theoretischer und empirischer) Überprüfung zugrunde liegender Hypothesen. Überprüft werden allerdings konsequent die Leistung und die Adäquatheit der Lösungen in Bezug auf angezielte Aufgaben – die wiederum keinen Gegenstand und kein Kriterium der reinen Linguistik darstellen.

Eine solche, für anwendungsorientierte Fachgebiete legitime Vorgehensweise ist dagegen bei Tätigkeiten mit wissenschaftlicher Zielsetzung und wissenschaftlichem Anspruch nicht akzeptabel. Es ist daher nicht nur sinnvoll, sondern unabdingbar zu prüfen, auf welchen theoretischen Grundlagen die (wissenschaftliche) Korpuslinguistik fußt bzw. fußen kann.

Die Erfahrung zeigt, dass Korpuslinguisten (wie Linguisten überhaupt) in aller Regel nicht über eine wissenschaftstheoretische Ausbildung verfügen. Dies wird nicht zuletzt an der unreflektierten Verwendung von Termini wie *Theorie* und *Erklärung* deutlich, die in der 'Mainstream'-Linguistik wie in der Computerlinguistik für formale Beschreibungs-

verfahren, Notationen, Begriffsdefinitionen, sogar Spekulationen u. v. m. verwendet werden[2].

Eine Sprachtheorie hat – wie jede Theorie – zum Ziel und ist in der Lage, die beobachteten und beschriebenen sprachlichen Phänomene zu *erklären* und neue, noch nicht beobachtete *vorherzusagen*. Es gibt keine Forschungsstrategie, die gegenüber anderen Ansätzen a priori eine größere Aussicht hat, dieses Ziel zu erreichen. In der heutigen Linguistik werden zwei verschiedene deduktive und mehrere induktiv-heuristische Strategien verfolgt.

Die zurzeit am weitesten verbreitete Form der deduktiven Sprachforschung besteht darin, unmittelbar Evidenz-gegründete Modelle der menschlichen Sprachfähigkeit und Sprachverarbeitung aufzustellen. Eine andere Strömung orientiert sich an den Naturwissenschaften und sucht universelle Sprachgesetze, die aus theoretischen Gründen für alle Sprachen und alle Zeiten gelten müssen. Diese Gesetze und die (psychologischen, physiologischen, physikalischen, soziologischen etc.) Randbedingungen schränken u. a. die Menge der möglichen Modelle ein. Die beiden Wege unterscheiden sich nicht hinsichtlich der empirischen Überprüfung: Aus den Modellen bzw. Gesetzen werden empirisch testbare Konsequenzen (Einzelhypothesen) abgeleitet und operationalisiert. Erst auf dieser Grundlage kann bestimmt werden, welche Art von Daten (Korpusdaten, experimentell erhobene Daten etc.) zu ihrer Falsifikation benötigt wird bzw. geeignet ist und wie diese zu interpretieren sind. Daten sind interpretierbar immer erst im Lichte einer Theorie oder wenigstens vor dem Hintergrund vortheoretischer Annahmen.

Wegen des immensen Aufwands der Aufbereitung und Interpretation großer Mengen sprachlicher Daten zur Hypothesenüberprüfung (für Bereiche zentralen Interesses sind sie bislang in nennenswertem Umfang nicht einmal möglich gewesen) sind Annotationen linguistischer Korpora wünschenswert, die für möglichst viele verschiedene Hypothesen aussagekräftig sind. Linguistisch annotierte Korpora, die der Forschung verfügbar sind, dienen also der Vermeidung von Doppelarbeit bei der Beschaffung von relevanten Daten und ermöglichen die Vergleichbarkeit und Replikation von Resultaten. Dennoch wird immer auch die Notwendigkeit für spezielle Korpusuntersuchungen bzw. für Annotationssysteme bestehen, die auf eine spezifische Fragestellung ausgerichtet sind.

Zur empirischen Überprüfung müssen linguistische Hypothesen in die Sprache der Statistik übersetzt werden, damit man sie mittels inferenzstatistischer Verfahren testen kann. Sie werden häufig z. B. in die Form von funktionalen Abhängigkeiten zwischen Variablen, in die Form von Frequenzverteilungen oder zeitabhängigen Entwicklungsgleichungen gebracht, aus Differential- bzw. Differenzengleichungen oder aus stochastischen Prozessen abgeleitet. Das Ergebnis der statistischen Analyse wird anschließend in die Sprache der Linguistik zurückübersetzt und führt entweder zur Ablehnung oder zur (vorläufigen) Beibehaltung der Hypothese.

Eine fundamentale Aufgabe jeder Wissenschaft ist die Schaffung einer Ordnung, das Finden von Mustern in der Menge mannigfaltiger, unübersichtlicher Daten. Klassifikations-, Korrelations-, Mustererkennungs- und andere induktiv-heuristische

[2] Wissenschaftstheoretische Grundbegriffe vermittelt die kurze Einführung in Altmann (1993). S. auch die dort angegebene weiterführende Literatur.

Verfahren dienen hauptsächlich dem Zweck, neue, zuvor nicht bekannte Phänomene und Zusammenhänge zu entdecken, zumal wenn, wie in der Korpuslinguistik, die Daten wegen ihrer schieren Masse mit dem Intellekt nicht einmal gesichtet werden könnten. Tatsächlich beruhen viele Erkenntnisse auf empirischen Generalisierungen, die nachträglich deduktiv verankert und ggf. modifiziert bzw. erweitert wurden.

Voraussetzungen für Fortschritte im Bereich der linguistischen Theoriebildung sind die anderen genannten Teilziele: die Verbesserung der methodologischen Grundlagen und die Erarbeitung einer adäquaten Datenbasis.

3 Methodische Grundlagen der Korpuslinguistik

Obwohl sich Linguisten und Computerlinguisten zunehmend mit korpuslinguistischen Fragestellungen beschäftigen, mangelt es bisher vielfach an Methodenbewusstsein. Dabei werden Bedingungen (wie z. B. Repräsentativität der Stichproben, die Homogenität der Daten und die Normalverteiltheit der Zufallsvariablen und der Abweichungen), die in anderen empirischen Wissenschaften meist automatisch als gegeben vorausgesetzt werden können, unberechtigterweise auch für linguistische Untersuchungen als erfüllt angesehen. Werden die besonderen statistischen Eigenschaften sprachlicher Daten berücksichtigt, ergeben sich grundlegende Vorbehalte gegen die unreflektierte Anwendung inferenzstatistischer Verfahren. Die wichtigsten der bis jetzt bekannten Probleme in diesem Bereich sind die folgenden:

(a) *Repräsentativität:* Keine Stichprobe kann repräsentative Sprachdaten in dem Sinne liefern, dass in dem in der Statistik üblichen Sinne gültige Schlussfolgerungen auf die Population, auf das "Sprachganze", möglich wären. Kein Korpus ist groß genug, um die Diversität der Daten im Hinblick auf Parameter wie Medium, Thematik, Stilebene, Genre, Textsorte, soziale, areale, dialektale Varietäten, gesprochene vs. geschriebene Texte etc. repräsentativ abzubilden. Versuche, das Problem durch Erweiterung der Stichprobe zu lösen, vergrößern nur die Diversität der Daten im Hinblick auf die bekannten (und möglicherweise noch unbekannte) Variabilitätsfaktoren und damit die Inhomogenität (s. Punkt b). Vor allem aber müsste es zur Beurteilung der Repräsentativität entweder theoretisches Vorwissen geben, aus dem die erforderlichen Mengenverhältnisse zwischen Texten mit den verschiedensten Eigenschaften hervorginge, oder ausreichende Erfahrungen mit unvorstellbar großen Textmengen aller denkbaren Arten, aus denen dann ein 'repräsentatives' Korpus eine Teilstichprobe wäre. Eine solch große Datensammlung ist aber nicht nur aus praktischen Gründen unmöglich, sondern auch, weil Sprache, Stile, Gesellschaften, Kulturen etc. nicht lange genug gleich bleibende Eigenschaften aufweisen, um hinlänglich viele gleichartige Daten entstehen zu lassen.

(b) *Die Homogenität der Daten:* Nur homogene Stichproben sind für viele der meistverwendeten statistischen Verfahren geeignet. Diese Bedingung ist für Sprachdaten nur selten erfüllt.

(c) *Die Normalverteiltheit der Zufallsvariablen und der Abweichungen:* Die wichtigsten Testverfahren, auf der eine Schlussfolgerung von der Stichprobe auf die Grundgesamtheit ja beruht, setzen voraus, dass die beobachteten Abweichungen von den erwarteten Werten der Zufallsvariablen normalverteilt sind. Diese Voraussetzung ist in der Sprache jedoch nicht generell erfüllt, so dass eigentlich für jeden einzelnen Fall gesonderte Tests abgeleitet werden müssten (eine mathematisch äußerst unbequeme und in der Praxis nicht durchführbare Forderung).

(d) *Die Homöoskedastizität:* Auch diese Bedingung, die gleichbleibende Varianz über alle Werte der betrachteten Zufallsvariablen, wird von Sprachdaten nicht generell erfüllt und muss besonders sorgfältig überprüft werden, bevor übliche Verfahren der Statistik angewendet werden dürfen.

(e) *Gültigkeitsbedingungen für Gesetzmäßigkeiten:* Von einigen Zusammenhängen und Gesetzen ist bereits bekannt, dass zu ihrer Erfüllung bestimmte Bedingungen erfüllt sein müssen. So kann im Gegensatz zu anderen Phänomenbereichen in der Sprache nicht von der Gültigkeit des Gesetzes der großen Zahlen ausgegangen werden. Ein anderes Beispiel für eingeschränkte Gültigkeitsbedingungen ist das bekannte Zipf-Mandelbrot-Gesetz, das nur für komplette Einzeltexte – nicht aber für Textfragmente oder Textkorpora gilt. Es ist zu vermuten, dass noch viele unbekannte Abhängigkeiten ähnlicher Art existieren, deren Kenntnis für korrekte Schlussfolgerungen unabdingbar wäre.

(f) *Die extreme Schiefe der Häufigkeitsverteilungen:* Dieses zentrale und für die Sprache typische Phänomen z. B. von Lauten, Silben, Wörtern (Formen und Bedeutungen) und syntaktischen Konstruktionen in Texten führt dazu, dass im Bereich der seltenen Einheiten stets – wie groß die analysierte Textbasis auch sei – eine nicht vernachlässigbare Unterrepräsentation vorliegt. Ein zweites Beispiel betrifft Stichproben aus Wörterbüchern oder Textvokabularen, die zwangsläufig eine Unterrepräsentation kurzer Wörter mit sich bringen[3].

(g) Direkte und indirekte funktionale Abhängigkeiten zwischen den linguistischen Größen wie Länge, Polysemie, Polytextie etc. bewirken, dass sich die entsprechenden Besonderheiten von Sprachdaten auf jede linguistische Untersuchung auswirken können. Dies gilt für Signifikanztests von Verteilungsanpassungen und Regressionen ebenso wie für Verfahren des Textvergleichs u. a.

Defizite in der Methodik sind auch deshalb zu beheben, weil der Zusammenhang zwischen Daten, den beobachtbaren Instanzen sprachlicher Äußerungen, und begründeten theoretischen Konstrukten im empirisch-induktiven Ansatz kompliziert und bislang nicht hinreichend geklärt ist. Für jede systematische Untersuchung von Korpora, die das empirische Wissen von Sprache vertiefen soll und dabei naturgemäß nicht ohne

[3] Zur Klarstellung sei betont, dass es an dieser Stelle nicht um 'Repräsentativität' von Korpora geht, sondern um die von Belegen einzelner, wohl definierter Eigenschaften.

theoretische Vorannahmen auskommt, und für jedes Untersuchungsziel müssen dabei Menge und Zulässigkeit der theoretischen Minimalannahmen geprüft werden, um die vorschnelle Festlegung der Befunde auf eine Bestätigung der Ausgangshypothese zu vermeiden.

Systematische Untersuchungen der allgemeinen statistischen Eigenschaften von Textkorpora im Sinne methodologischer Grundlagenforschung sind ferner erforderlich, um verlässliche Verfahren zur Eignungs- und Qualitätssicherung der Daten bei gegebener Anwendung bereitzustellen.

Die linguistische Untersuchung von empirischen Sprachdaten mit quantitativen mathematischen Mitteln hat eine lange, vor allem europäische Tradition, die in den USA unter dem dominanten Einfluss der formalen und Kompetenz-orientierten Linguistik jedoch kaum rezipiert wurde. Im Gegensatz dazu blieben die quantitativen Modelle und Verfahren in Russland und vielen mittel- und osteuropäischen Ländern immer selbstverständlicher Bestandteil des sprachwissenschaftlichen Instrumentariums. Die Entwicklung wissenschaftlicher Methoden für deskriptive Zwecke ist mit Namen wie Zipf (z. B. 1949, 1968), Herdan (z. B. 1966), Menzerath (z. B. 1954), Tuldava (z. B. 1995, 1998) und Piotrowski (z. B. 1984); Piotrowski et al. (z. B. 1985) verknüpft. Für das Vordringen in eine explanative Phase ist vor allem das Pionierwerk von Gabriel Altmann von größter Bedeutung; es bietet eine ausgezeichnete Grundlage in Hinblick auf die wissenschaftstheoretische (epistemologische und methodologische) Reflexion und Fundierung der linguistischen Forschung und liefert fundamentale Beiträge zur mathematischen Modellbildung, zur Theoriebildung durch die Formulierung einer Reihe von universellen Sprach- und Textgesetzen und zur quantitativ-linguistischen Methodik (s. z. B. Altmann, 1981, 1988, 1993, 1995; Altmann und Schwibbe, 1989; Altmann und Hřebíček, 1993, u. v. m.). Auf dieser Basis entstand auch der integrative systemtheoretische Modellrahmen der "synergetischen Linguistik" (vgl. z. B. Köhler, 1986, 1987, 1999). In jüngerer Zeit haben sich nicht nur viele Forscher dieser Strömung geöffnet, sondern es gibt sogar eine zunehmende Tendenz dazu, linguistische Fortschritte vor allem aus dieser Richtung zu erwarten. Seit einigen Jahren werden quantitative Hilfsmittel verstärkt auch in den USA aufgegriffen (s. z. B. Church, Mercer, IBM), von wo aus wiederum eine intensivierende Rückwirkung nach Europa zu verspüren ist. Für diesen ganzen Bereich vgl. vor allem auch das aktuelle Handbuch (Köhler, Altmann und Piotrowski, 2005) und die Bibliographie (Köhler, 1995).

4 Verbesserung der Korpustechnik und der Ressourcennutzung

Die Entwicklung von Korpora ist selbst dann zeitaufwändig und kostenintensiv, wenn diese nach opportunistischen Kriterien wohlstrukturiert aufgebaut wurden (Übernahme jeder Art von maschinenlesbarem, kostenlos verfügbarem Text bei geklärten Nutzungsrechten) und nicht anwendungsorientiert, im Sinne von zulässigen und notwendigen Vorannahmen. Korpora, die dem jeweiligen Untersuchungsziel angemessen sind und deren Zusammensetzung linguistisch begründet ist, sind demnach in der Entwicklung noch erheblich kostspieliger und zeitaufwändiger. Dazu gehören beispielsweise parallele

Korpora, deren Textelemente Übersetzungen voneinander sind, und, da Korpora "altern", auch dynamische, ständig durch neue, bislang unbeobachtete Phänomene ergänzte Korpora (sogenannte "Monitorkorpora"). Die Größe solcher in Universitäten verfügbarer Korpora schwankt heutzutage zwischen minimal 1 Million Wörter und ca. 100 Millionen Wörter, erreicht in Ausnahmefällen jedoch auch erheblich größere Zahlen.

Für die datenorientierte Linguistik sind neben umfangreichen textuellen auch lexikale Daten und Wörterbuchressourcen von größter Bedeutung, da sie als Hilfsmittel für nicht triviale Korpusauswertungsverfahren benötigt werden. Dazu gehören z. B. monolinguale Frequenzwörterbücher, Trivia (aus Sicht der Theorie) wie umfassende Abkürzungs- und Namenslisten, Thesauri, semantische Wortnetze, Valenzwörterbücher, bilinguale Wörterbücher etc.

Die für Korpora und lexikale Ressourcen erforderlichen hohen Aufwendungen stehen einer breiten Nutzung empirischer Daten entgegen und erschweren sogar den Zugang zu existierenden Sammlungen, da häufig fremde Daten nur im Tausch zugänglich gemacht werden. Obwohl inzwischen in Deutschland an mehreren Stellen unterschiedlichste Korpora und zum Teil auch lexikale Ressourcen existieren, liegen zur Zeit keine zuverlässigen, aktuellen und vollständigen Informationen darüber vor. Es wäre also anzustreben, die gegenseitige Information über vorhandene Daten und die Erleichterung des Zugangs zu ihnen zu verbessern.

Gleiches gilt für die Information über Softwarewerkzeuge zum Aufbereiten von Rohdaten ("text encoding") zwecks Standardisierung von Austauschformaten, für Software zum automatischen Annotieren der Daten bis hin zu Parsern und Werkbänken für die interaktive grammatische Analyse und Paketen für die statistischen Analysen. Zuverlässige Informationen und die Erleichterung des Zugangs zu Werkzeugen sind trotz durchaus beobachtbarer Bemühungen noch nicht ausreichend gegeben.

4.1 Korpus-Standardisierung

Der Aufwand, der für die Erstellung und Wartung von Korpora betrieben werden muss, rechtfertigt einige zusätzliche Gedanken und auch eine gewisse Zusatzinvestition (in Form von Strukturierung und Programmierung), um den Gesamtnutzen zu maximieren: Gegenwärtig halten die meisten Computer- und Korpuslinguisten das Problem der Standardisierung von Datenrepräsentationen und -schnittstellen mit der Verfügbarkeit von Auszeichnungssprachen wie SGML und XML und von Werkzeugen zu ihrer problemlosen Nutzung für gelöst. Dies ist jedoch ein Irrtum. So sehr diese Möglichkeiten einen echten Fortschritt darstellen – sie bilden nur eine Notationsmöglichkeit. Worin die tieferen Probleme liegen, sollen die folgenden Überlegungen zeigen:

1. Auch die *Verwendung* eines Korpus ist mit Überlegungen und Arbeit verbunden, selbst wenn das Korpus fertig vorgefunden wird; dieser Aufwand für die Korpus-Nutzung sollte möglichst minimiert werden;

2. Es ist äußerst ineffizient, für jede Untersuchung, welche die Nutzung eines Korpus einbezieht, alle diese Überlegungen und Arbeiten von Neuem durchführen zu

müssen, nur weil irgendwelche Details in der Aufbereitung oder der Organisation des verwendeten Korpus nicht zu der intendierten Untersuchung passen.

Ein anderes häufiges aber gleichwohl wenig beachtetes Problem ist das der suboptimalen Bewahrung der Originaldaten (auch kurz: Informationsvernichtung). Als illustrierendes Beispiel kann z. B. ein Linguist dienen, der Zugang zu den Satzbändern einer Tageszeitung hat. Er verwendet diese Bänder, um aus ihnen ein Korpus aus Zeitungstexten zu erstellen. Außer dem eigentlichen Text sind auf diesen Bändern noch eine Menge "merkwürdiger" Steuerzeichen enthalten, welche die Satzmaschinen steuern und mit der Positionierung und Gestaltung der Texte zu tun haben. In der Regel wird unser Linguist sorgfältig bemüht sein, diese "nutzlosen und störenden" Sequenzen aus dem Datenstrom zu entfernen. Eine Konsequenz dieser verbreiteten Vorgehensweise ist, dass andere Forscher, z. B. Inhaltsanalytiker, die für ihre Fragestellungen gerade die Information über Position und Größe der Aufmachung benötigen würden – also exakt die Information, die in den "merkwürdigen, nutzlosen und störenden" Zeichen verborgen war – das Korpus nicht verwenden können.

Im Nachhinein betrachtet kann man den beschriebenen Vorgang kaum verstehen: Viel Mühe wurde aufgewendet mit dem Resultat, dass wertvolle Daten zerstört wurden. Andererseits wird man zwei Dinge zugeben müssen:

1. Unser Beispiel-Linguist hatte nicht die geringste Idee, dass die von ihm entfernten Zeichenfolgen von irgend einem Interesse sein könnten, und wenn er sie gehabt hätte, hätte er nicht gewusst, ob tatsächlich irgendwann jemand an seinem Korpus Interesse gezeigt hätte;

2. Die vereinfachte Form seines Korpus ist erheblich transparenter und effizienter im Hinblick auf die Verarbeitung zu seinen eigenen Zwecken. So müssen die Auswertungsprogramme sich nicht um die möglicher Weise komplizierten technischen Details kümmern, die ohnehin zu der bezweckten Untersuchung nichts beitragen.

Allerdings gilt allgemein: Je mehr ein Korpus für einen bestimmten Zweck optimiert wurde, desto schwieriger wird es, es für einen anderen Zweck zu verwenden. Eine einfache Methode, dieses Problem zu beheben, besteht darin, die zunächst nicht benötigten Daten zu kapseln, also mit einer entsprechenden Kommentierung zu klammern, so dass sie überlesen werden können.

Selbstverständlich ist dieses Beispiel extrem. Die meisten der in der Korpuslinguistik diskutierten technischen Themen betreffen viel weniger spektakuläre Fragen, darunter Erörterungen über die Auswahl und Verwendung der jeweils populären Auszeichnungssprachen (wie eben SGML, HTML, XML etc.), die Entscheidung für eines der prominenten Wortklassen-Tagsets, Vor- und Nachteile von Dokumentrepräsentationssystemen (PDF) und viele andere. Zu bedenken ist auch, dass es eine Vielzahl von Formaten (Dokumentenstrukturen) gibt, in denen die Texte dargestellt werden können: reiner, laufender Text mit Texttrennern, annotierter Text (z. B. mit Wortklassenzuordnung, syntaktische Analysebäume in Form etikettierter Klammergebirge oder in Form eingerückter Zeilen

mit Marken, Dateien mit einer Zeilenstruktur, bei denen jede Zeile ein Textwort mit einer Reihe verschiedener Annotate enthält, Dateien mit reinem Text in Begleitung separater Annotationsdateien, aus denen Zeiger von den Annotaten auf die referenzierten Einheiten der Texte verweisen etc.). Die Auswahl unter den Möglichkeiten wird man natürlich aufgrund der gegebenen Umstände und des Verwendungszwecks treffen.

Darüber hinaus ist zu beachten, dass jedes Korpus bestimmte technische Merkmale besitzt, die oft nicht völlig in der Entscheidung der Korpus-Ersteller liegen: Betriebssysteme, Dateisysteme, Zeichencodes (wie ASCII, EBCDIC, Unicode, um einige der zurzeit bekanntesten zu nennen), Massenspeichertypen, Zugriffsmethoden (ein Korpus kann aus einer einzigen großen Datei bestehen oder aus Tausenden von Einzeldateien, es kann über mehrere Rechner in einem Netzwerk verteilt oder auf einer einzigen CD-ROM gespeichert sein. Die technische Repräsentation kann sich sogar dynamisch verändern; man bedenke auch, dass die Lebensdauer eines guten Korpus als deutlich höher veranschlagt werden sollte als die von Speichermedien, Betriebssystemen, Zeichencodes und Darstellungssprachen.)

Was selten bedacht wird ist, dass nahezu jede denkbare Kombination von Korpusmerkmalen und ihren Ausprägungen realisiert sein kann. Benutzer von Korpora und Programmierer von Analyse- oder Bearbeitungssoftware, die mit mehr als einem einzigen, speziellen Korpus arbeiten können soll, sind mit einem riesigen Spektrum von Strukturen und technischen Einzelheiten konfrontiert: Jedes Korpus ist ein Spezialfall, auch wenn es – um das zu wiederholen – mit XML aufgezeichnet wurde.

Von der anderen Seite her betrachtet wird es noch unangenehmer: Wenn auch nur in einem einzigen Korpus ein Detail verändert wird (was durchaus nötig werden kann, auch wenn die Vorüberlegungen sehr gründlich waren), müssen alle Programme, die mit diesem Korpus arbeiten sollen, angepasst werden.

Überraschender Weise wird allen diesen mit riesigem Aufwand behafteten Problemen kaum Aufmerksamkeit geschenkt, obwohl die Softwaretechnik Standardlösungen für sie bereit stellt.

4.2 Abstrakte Datenstrukturen und abstrakte Datentypen

Betrachten wir zur Einführung ein sehr einfaches Beispiel: die Programmieraufgabe, zwei Zahlen miteinander zu addieren. Für diese Aufgabe war es in den Anfangszeiten der Rechnertechnik erforderlich, genau zu wissen, wo im Speicher des Computers (z. B. in welchem Register, welcher Indexzelle oder unter welcher Adresse im Kernspeicher) diese Zahlen zu finden waren und auf welche Weise sie in dem betreffenden Computer repräsentiert waren (z. B. vier Bytes für die Mantisse und ein Byte für den Exponenten in einer bestimmten Reihenfolge, wobei zwei der Bits als Vorzeichen von Mantisse bzw. Exponent, andere zur Fehlererkennung etc. dienen können, und noch klar sein musste, welches der beiden Nibbles eines Bytes (ein Nibble besteht aus 4 Bits) als das obere bzw. untere zu gelten hatte; zudem musste die Adressierungsart von Bytes und/oder Maschinenwörtern bekannt sein u. v. m.). Ohne die Kenntnis all dieser Einzelheiten wäre

es unmöglich gewesen, ein Programm(stück) auch nur zum Addieren zweier Zahlen zu schreiben.

Später, mit der Einführung von Programmiersprachen, wurde diese Aufgabe erheblich erleichtert. Programmiersprachen stellen Operatoren wie die Addition (meist mit dem Zeichen '+' symbolisiert) zur Verfügung, die verwendet werden können, ohne Einzelheiten der Implementierung und Speicherung der Operanden zu kennen. Es ist ein gutes Designprinzip einer Programmiersprache, solche Implementierungsdetails (wie auch die Arbeitsweise der Algorithmen, welche die Operatoren realisieren) vor dem Programmierer sogar zu verbergen. In der Softwaretechnik ist dieses Prinzip unter der Bezeichnung Geheimnisprinzip ("information hiding") bekannt, und viele gute Gründe sprechen für die strikte Einhaltung dieses Prinzips. Die beiden wichtigsten sind die folgenden:

1. Wenn man bei der Programmierung die Details nicht kennen (und somit berücksichtigen) muss, weil die Programmiersprache selbst dafür sorgt, dann kann das entstandene Programm auf allen existierenden und zukünftigen Computeranlagen der Welt und unter allen denkbaren Betriebssystemen etc. laufen, unter denen die betreffende Programmiersprache verfügbar ist.

2. Das Geheimnisprinzip hindert den Programmierer daran, die Kenntnis von Repräsentationen, Arbeitsweisen und anderen technischen Details in seinem Programm auszunutzen, was in einer veränderten Umgebung (Hardware, Betriebssystemversion etc.) zu fehlerhaftem Verhalten oder Abstürzen führen würde.

Eine weitere Verbesserung der Programmiertechnik entstand durch die Einführung von Datentypen in den Programmiersprachen, die dafür sorgen, dass die Programmierer nicht Äpfel mit Birnen vergleichen oder eine Zahl mit einem Buchstaben multiplizieren (können). Jeder Operator ist im Hinblick auf seine möglichen Operanden (Argumente) und auf die Eigenschaften des Ergebnisses definiert. Moderne Programmiersprachen erlauben die Definition eigener Operatoren, meist in Form von Funktionen und Prozeduren.

Die Verwendung von Funktionen und Prozeduren führt auch zu einer verbesserten Programmstruktur (Lesbarkeit, Veränderbarkeit, Portierbarkeit, Wartbarkeit und andere Gütekriterien der Softwareentwicklung). Dies ist bei der Ausbildung von Programmierern ebenso zu betonen wie die Vorteile wiederverwendbarer Software. Eine Prozedur, die z. B. zur Suche des Maximums in einer Liste von Zahlen oder zur Sortierung einer Liste nach einem gegebenen Kriterium geschrieben wurde, kann nicht nur innerhalb des Programms verwendet werden, für das sie ursprünglich geschrieben wurde, sondern auch in unzähligen weiteren Programmen, in denen ähnliche Aufgaben vorkommen – wenn die betreffende Prozedur allgemein genug formuliert wurde.

Wiederverwendbarkeit ist das Hauptanliegen abstrakter Datenstrukturen (ADS) und abstrakter Datentypen (ADT), die noch einen Schritt weiter gehen als gewöhnliche (vordefinierte) Datentypen: Sie versetzen den Programmierer in die Lage, darüber hinaus eigene Datentypen mit dazugehörigen Operatoren zu kreieren. Das Besondere an ADS

und ADT ist, dass ihre Implementierungsdetails dennoch verborgen werden: Sie bestehen aus einem Datenobjekt mit den erforderlichen Zugangsprozeduren im Fall der ADS und aus einer Klasse von Objekten im Fall der ADT. Letztere erlauben die Schaffung von mehr als einer Variablen des gegebenen Datentyps während der Laufzeit. Betrachten wir das folgende Beispiel. Viele (auch komplexe) Datenstrukturen kommen extrem häufig vor; doch im Rahmen der herkömmlichen Programmiertechnik schreibt jeder Programmierer seinen eigenen Code für eine Liste oder eine Matrix, einen Stack oder einen Baum – jedes Mal, wenn er eine solche Struktur benötigt (er wird, natürlich, so viel wie möglich von vorherigen eigenen oder z. B. aus dem Internet erhältlichen fremden Implementierungen kopieren und wird dabei, natürlich, Fehler machen). Wesentlich bei ADS und ADT ist die Realisierung der zu den Strukturen gehörenden Mechanismen in einer von dem jeweiligen, konkreten Problem unabhängigen, allgemeinen Weise, d. h. ohne Berücksichtigung der konkreten Verwendung z. B. eines Stacks in einem Parser, Compiler oder Suchprogramm. Was zählt ist, dass ein Stack einen Konstruktor (dargestellt als `CREATE`, `NEW` o. ä.), Modifikatoren (wie `PUSH` oder `POP`) und Inspektoren (wie `TOP` und `EMPTY`) und deren Wirkung auf die Daten definiert. Der Benutzer eines Stacks muss nicht und sollte nicht wissen, wie die entsprechenden Funktionen und Prozeduren arbeiten oder wie die Datenstruktur implementiert wurde (z. B. in Form einer einfach oder doppelt verketteten Liste mit Zeiger oder auch nur als Feld (`array`) – so wie ein guter Programmierer die Elemente einer Programmiersprache verwendet, ohne zu berücksichtigen, wie die Datentypen array, set, real oder boolean jeweils implementiert sind. Er braucht nur die Kenntnis der zu den Operationen gehörenden Vor- und Nachbedingungen. Im Beispiel des Stacks hat der Konstruktor `CREATE` keine Vorbedingung (ein neuer Stack kann jederzeit kreiert werden); seine Nachbedingung ist, dass `EMPTY` den Wert `TRUE` hat. Der Modifikator `PUSH(x)` hat die Vorbedingung, dass der Stack existiert. Seine Nachbedingung ist, dass `TOP` den Wert `x` besitzt. Ein Modul, das eine Datenstruktur (deren interner Aufbau verborgen bleibt) zusammen mit den notwendigen Zugriffsprozeduren (Konstruktoren, Modifikatoren, Inspektoren, deren Arbeitsweise im Einzelnen ebenfalls verheimlicht wird) realisiert, nennt man auch Datenkapsel.

4.3 Textkorpora als ADS

Offensichtlich können die dargestellten Prinzipien der Softwaretechnik auf die im ersten Abschnitt diskutierten Probleme angewendet werden. Die Situation eines Programmierers, der zwei Zahlen zu addieren hat (und nicht notwendigerweise die binäre oder die BCD-Addition neu erfinden möchte/sollte), kann mit der des Korpus-Anwenders verglichen werden, der in einer programmierten Schleife Silbe für Silbe, Wort für Wort oder Satz für Satz auf ihn interessierende Merkmale untersuchen will (und nicht wirklich daran interessiert ist herauszufinden, wie man in einem gegebenen Korpus die jeweils nächste Einheit zweifelsfrei zu finden, zu identifizieren und zu segmentieren hat). Alle Eigenschaften und Einzelheiten, die spezifisch für ein Korpus sind, sollten daher gekapselt werden, während das Korpus und seine Inhalte dem Benutzer auf einer Ebene präsentiert werden sollte, die seinen Interessen entsprechen – so wie höhere Programmiersprachen

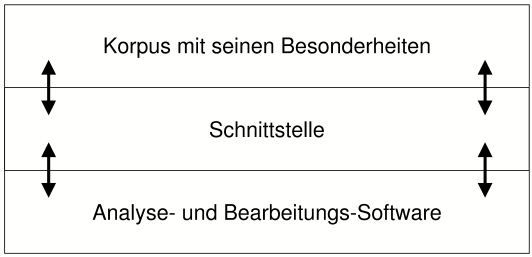

Abbildung 1: Das Prinzip der Korpus-Schnittstelle.

von technischen Details abstrahieren und dem Programmierer Werkzeuge auf der Ebene seines Problems anbieten (vgl. Abb. 1). Dies bedeutet auch, dass es für ein Korpus mehr als eine Darstellung oder eine Schnittstelle geben kann. So sollte die Schnittstelle Befehle wie "Gib mir die nächste Silbe" oder "Gib mir die Wortart der aktuell betrachteten Wortform" in Prozeduren übersetzen, die gerade dies tun. Dazu muss die Schnittstelle (nicht aber der Nutzer bzw. das nutzende Programm) über die Kenntnisse verfügen, wie in dem jeweiligen Korpus Silben repräsentiert sind und wie die jeweils nächste aufzufinden ist, während die auf die Schnittstelle zugreifende Software entsprechend problemnah formuliert werden kann.

Allgemein sollte die Schnittstelle in der Lage sein, alle für Untersuchungen interessanten Einheiten, Kategorien, Eigenschaften (Annotate) etc. aus dem Korpus an das nutzende Programm zu liefern. Auf der Zeichenebene sollten neben Buchstaben(folgen) die Separatoren und Interpunktionen abrufbar sein, ebenso Information über die Schreibweise (Groß-/Kleinbuchstaben, Schriftart, Auszeichnungen), die Position des Strings (relativ zum Text, Absatz, Satz, …), kurz alles, was explizit im Korpus annotiert wurde oder für die Schnittstellensoftware leicht erkennbar oder erschließbar (z. B. die Länge von Einheiten) ist. Ähnlich sollten auf den Silben-, Morph(em)-, Wort-, Phrasen-, Satz- etc. -ebenen zusammen mit den jeweiligen Einheiten selbst alle typographischen, linguistischen und sonstigen Informationen als Wert eines komplexen Prozedurparameters übergeben werden.

Das Korpus-Interface sollte bidirektional ausgelegt werden; es sollte also auch Prozeduren (Konstruktoren und Modifikatoren) zur Verfügung stellen, die das Annotieren und andere Bearbeitungen erlauben, vorausgesetzt, das Programm, das die Schnittstelle verwendet, besitzt die dazu erforderlichen Rechte. So hätte das bearbeitende Programm (ein interaktiver Editor für die manuelle Annotation ebenso wie ein automatischer Tagger oder Parser) keinerlei Information darüber, in welcher Weise die Annotationen gespeichert würden (genauso wie dies beim Lesen des Korpus und seiner Annotationen unbekannt bleibt).

Eine wichtige Grundfunktion der Schnittstelle realisiert diejenige Prozedur, die dem aufrufenden Programm Auskunft darüber erteilt, welche Möglichkeiten, Kategorien,

Elemente und Annotationen in der gegebenen Korpusversion mit der gegebenen Schnittstellenversion verfügbar sind. Dazu gehören auch Informationen über das verwendete Alphabet, Sonderzeichen, erlaubte Parameterwerte, Einschränkungen usw.

Schließlich stellt sich die Frage, woher die Schnittstelle selbst all die genannten und vielleicht noch viele weitere Informationen erhält. Selbstverständlich sollten diese Dinge nicht fest in der Schnittstellensoftware kodiert werden. Die Nachteile einer solchen Lösung sind offensichtlich: Die Schnittstellensoftware müsste für jedes einzelne Korpus und auch nach jeder Änderung auch nur eines Korpus angepasst und rekompiliert werden. Außerdem würde das dazu führen, dass zahlreiche Versionen der Schnittstelle entstehen, von denen jeweils nur eine mit jedem Korpus arbeiten könnte. Die falschen Versionen würden nur Fehlermeldungen produzieren oder, schlimmer, unerkannt falsche Ergebnisse liefern.

Also wird eine unabhängige Korpusbeschreibung benötigt: eine Datei, die alle erforderlichen Informationen über das Korpus enthält, einschließlich der Auskünfte darüber, wo sich das Korpus (oder seine Teile) befindet und wie darauf zuzugreifen ist. Der beste Weg, das Korpus für das Schnittstellen-Modul zu beschreiben, ist die Verwendung einer formalen Sprache, am besten einer LL(1)-Sprache. Solche Sprachen besitzen Eigenschaften, die sie für einen Parser besonders leicht zu verarbeiten machen (cf. Aho et al., 1988; Rechenberg und Mössenböck, 1985; Wirth, 1986). Diese Beschreibung muss vom Korpus-Ersteller zur Verfügung gestellt werden. Die allgemeine Architektur einer Korpus-Schnittstelle, wie sie hier vorgeschlagen wird, ist aus der Abbildung 2 (am Ende dieses Beitrags) ersichtlich.

5 Schluss

Dieser Beitrag hat versucht, einige wesentliche Defizite aufzuzeigen, welche die heutige Korpuslinguistik aufweist, ohne zu verkennen, dass sie einige dieser Defizite mit anderen Teildisziplinen teilt. Es sollten auch weder die bereits erzielten Fortschritte noch die Verdienste der Korpuslinguistik bestritten werden. Vielmehr soll der Beitrag als konstruktive Kritik verstanden werden, die auch Perspektiven und aussichtsreiche Ansätze zur Überwindung der genannten Defizite zeigt.

Literatur

Aho, A. V., R. Sethi und J. D. Ullman (1988). *Compilers: principles, techniques, and tools*. Reading, Massachusetts: Addison-Wesley.

Altmann, G. (1981). Zur Funktionalanalyse in der Linguistik. In J. Esser und A. Hübler (Hrsg.), *Forms and Functions: Papers in General, English & Applied Linguistics Presented to Vilem Fried on the Occasion of His Sixty-Fifth Birthday*, Band 149, *Tübinger Beiträge zur Linguistik*, S. 25–32. Tübingen: Narr.

Altmann, G. (1988). *Wiederholungen in Texten*. Bochum: Brockmeyer.

Altmann, G. (1993). Science and linguistics. In R. Köhler und B. B. Rieger (Hrsg.), *Contributions to Quantitative Linguistics*, S. 3–10. Dordrecht: Kluwer.

Altmann, G. (1995). *Statistik für Linguisten*. Trier: Wissenschaftlicher Verlag Trier.

Altmann, G. und L. Hřebíček (Hrsg.) (1993). *Quantitative Text Analysis*. Trier: Wissenschaftlicher Verlag Trier.

Altmann, G. und M. H. Schwibbe (1989). *Das Menzerathsche Gesetz in informationsverarbeitenden Systemen*. Hildesheim: Georg Olms.

Herdan, G. (1966). *The Advanced Theory of Language as Choice and Chance*. Berlin/Heidelberg/New York: Springer.

Köhler, R. (1986). *Zur linguistischen Synergetik. Struktur und Dynamik der Lexik*. Bochum: Brockmeyer.

Köhler, R. (1987). Systems theoretical linguistics. *Theoretical Linguistics 14*(2/3), 241–257.

Köhler, R. (1995). *Bibliography of Quantitative Linguistics (Bibliographie der quantitativen Linguistik; Библиография по квантитативной лингвистике)*. Amsterdam/Philadelphia: Benjamins.

Köhler, R. (1999). Syntactic Structures. Properties and Interrelations. *Journal of Quantitative Linguistics 6*, 46–57.

Köhler, R., G. Altmann und R. G. Piotrowski (Hrsg.) (2005). *Quantitative Linguistik. Ein internationales Handbuch. / Quantitative Linguistics. An International Handbook*. Berlin/New York: de Gruyter.

Köhler, R. und B. B. Rieger (Hrsg.) (1993). *Contributions to Quantitative Linguistics. Proceedings of the First Quantitative Linguistics Conference (QUALICO-91)*. Dordrecht: Kluwer.

Menzerath, P. (1954). *Die Architektonik des deutschen Wortschatzes*. Bonn: Dümmler.

Piotrowski, R. G. (1984). *Inženernaja lingvistika i teorija jazyka*. Leningrad.

Piotrowski, R. G., K. Bektaev und A. Piotrowskaja (1985). *Mathematische Linguistik*. Bochum: Brockmeyer.

Rechenberg, P. und H. Mössenböck (1985). *Ein Compiler-Generator für Mikrocomputer. Grundlagen. Anwendung. Programmierung in Modula-2*. München: Hanser.

Tuldava, J. (1995). *Methods in quantitative linguistics*. Trier: Wissenschaftlicher Verlag Trier.

Tuldava, J. (1998). *Probleme und Methoden der quantitativ-systemischen Lexikologie [übersetzte, verbesserte und ergänzte Fassung von: Problemy i metody kvantitativno-sistemnogo issledovanija leksiki, 1987]*. Trier: Wissenschaftlicher Verlag Trier.

Wirth, N. (1986). *Compilerbau. Eine Einführung* (4 Aufl.). Stuttgart: Teubner.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort. An Introduction to Human Ecology*. Reading, Massachusetts: Addison-Wesley.

Zipf, G. K. (1968). *The Psycho-Biology of Language. An Introduction to dynamic philology*. Cambridge, Massachusetts: MIT Press.

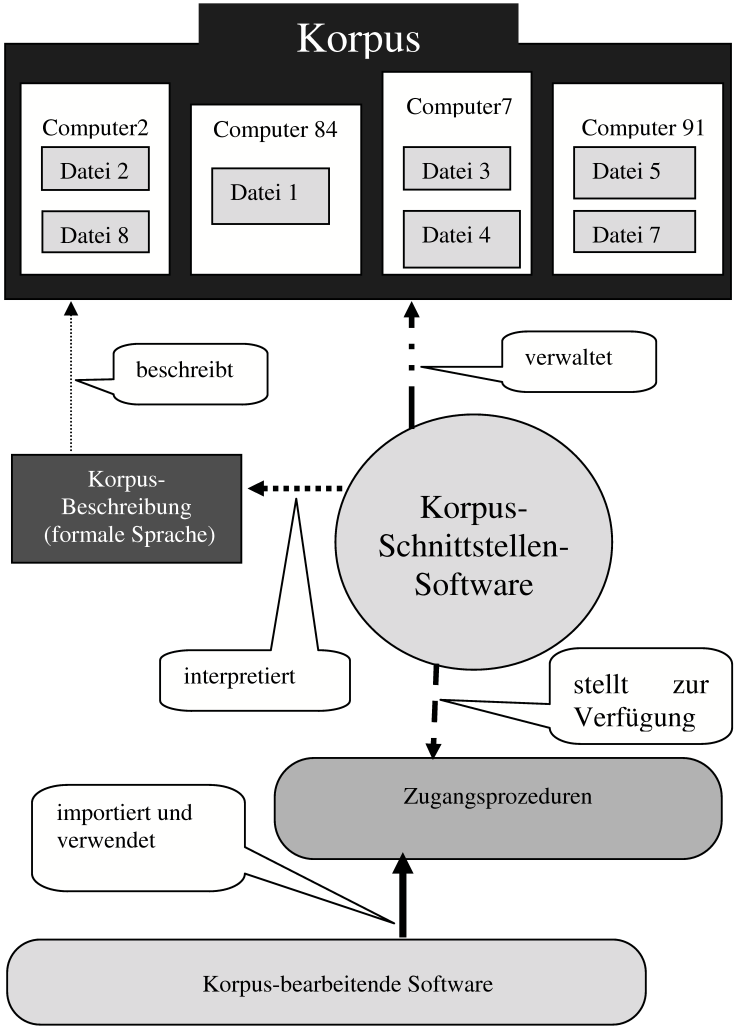

Abbildung 2: Eine Architektur einer allgemeinen Korpus-Schnittstelle.