Band 20 – Heft 2 – Jahrgang 2005 – ISSN 0175-1336
Zeitschrift für Computerlinguistik und Sprachtechnologie
GLDV-Journal for Computational Linguistics and Language Technology

LDV Forum

Themenschwerpunkt
Korpuslinguistik

Herausgegeben von
Alexander Mehler

Gesellschaft für linguistische Datenverarbeitung www.gldv.org

Korpuslinguistik

LDV/ Impressum

LDV-Forum Zeitschrift für Computerlinguistik und Sprachtechnologie
ISSN 0175-1336 GLDV-Journal for Computational Linguistics and Language
Band 20 - 2005 - Heft 2 Technology – Offizielles Organ der GLDV

Herausgeber Gesellschaft für Linguistische Datenverarbeitung e. V. (GLDV)

Juniorprofessor Dr. Alexander Mehler, Universität Bielefeld,
alexander.mehler@uni-bielefeld.de
Prof. Dr. Christian Wolff, Universität Regensburg
christian.wolff@sprachlit.uni-regensburg.de

Anschrift der Redaktion Prof. Dr. Christian Wolff,
Universität Regensburg
Institut für Medien-, Informations- und Kulturwissenschaft
D-93040 Regensburg

Wissenschaftlicher Beirat Vorstand, Beirat und Arbeitskreisleiter der GLDV
http://www.gldv.org/cms/vorstand.php,
http://www.gldv.org/cms/topics.php

Erscheinungsweise 2 Hefte im Jahr, halbjährlich zum 31. Mai und 31. Oktober. Preprints und redaktionelle Planungen sind über die Website der GLDV einsehbar (*http://www.gldv.org*).

Einreichung von Beiträgen Unaufgefordert eingesandte Fachbeiträge werden vor Veröffentlichung von mindestens zwei ReferentInnen begutachtet. Manuskripte sollten deshalb möglichst frühzeitig eingereicht werden und bei Annahme zur Veröffentlichung in jedem Fall elektronisch und zusätzlich auf Papier übermittelt werden. Die namentlich gezeichneten Beiträge geben ausschließlich die Meinung der AutorInnen wieder. Einreichungen sind an die Herausgeber zu übermitteln.

Bezugsbedingungen Für Mitglieder der GLDV ist der Bezugspreis des LDV-Forums im Jahresbeitrag mit eingeschlossen. Jahresabonnements können zum Preis von 25,- € (inkl. Versand), Einzelexemplare zum Preis von 15,- € (zzgl. Versandkosten) bei der Redaktion bestellt werden.

Satz und Druck Carolin Kram, Bielefeld, mit *LaTeX* (*pdfeTeX* / *pdfLaTeX* / *MiKTeX*) und *Adobe InDesign CS 3.0.1* , Druck: Druck TEAM KG, Regensburg

Alexander Mehler und Christian Wolff

Editorial

Liebe GLDV-Mitglieder, liebe Leserinnen und Leser des LDV-Forums,

mit dem Erscheinen dieses Hefts wird der reguläre Publikationszyklus des LDV-Forums für dieses Jahr abgeschlossen. Nachdem im Frühjahr das Themenheft *Text Mining* erschienen ist, fokussiert die vorliegende Ausgabe auf den Bereich der Korpuslinguistik, und zwar unter texttechnologischer, quantitativer und computerlinguistischer Perspektive. Dies betrifft methodologische Fragestellungen nach den Möglichkeiten und Grenzen statistischer Korpusanalysen, die Erstellung und Nutzung von Korpora nach Prinzipien des *Software Engineering* (Reinhard Köhler), die Standardisierung von Ressourcen für die Sprachverarbeitung (Thorsten Trippel, Thierry Declerck und Ulrich Heid), die Kategorisierung textueller Einheiten auf der Grundlage quantitativer Stilcharakteristika (Emmerich Kelih und Peter Grzybek), die Aufbereitung Webgenre-spezifischer HTML-Seiten für korpuslinguistische Untersuchungen (Georg Rehm) sowie die korpusbasierte Exploration von Ontologien (Chris Biemann). Damit decken die Beiträge ein Spektrum ab, das von der Korpuserstellung über die statistische Korpusanalyse bis hin zur computerlinguistischen Modellierung reicht, wobei der Beitrag zum Ontologie-Lernen an das vorangehende Themenheft anknüpft.

Dass die vorliegende Ausgabe ebenfalls in Form eines Themenschwerpunkthefts erscheint, bedeutet jedoch keine Festlegung für nachfolgende Ausgaben. Es drückt sich darin vielmehr die Bedeutung korpuslinguistischer Fragestellungen im Zusammenhang computerlinguistischer und texttechnologischer Entwicklungen aus, die im Verbund auf die automatische Analyse sprachlicher Daten zielen, und zwar mittels Prozeduren, deren Input- und Outputeinheiten auf der Grundlage texttechnologischer Prinzipien modelliert und ausgezeichnet werden.

Für das Jahr 2006 sind zwei weitere Ausgaben des Forums geplant, wobei für die Herbstausgabe ein *Call for Papers* in Vorbereitung ist, der unter anderem über den GLDV-Verteiler veröffentlicht werden wird. Darüber hinaus freuen wir uns, auf ein laufendes Projekt verweisen zu können, dass die Digitalisierung der zurückliegenden Ausgaben des Forums beinhaltet. Ziel ist es, die digitalisierten Ausgaben über das GLDV-Portal frei zugänglich zu machen. Auf diese Weise hoffen wir, den Status des Forums auch als elektronische Zeitschrift nachhaltig stärken zu können. An dieser Stelle rufen wir die Autorinnen und Autoren von Forumsbeiträgen aus der Zeit vor 1997 dazu auf, uns Ihre elektronischen Fassungen (z.B. in LaTeX, Word, PDF oder PS) zukommen zulassen (Kontaktadresse: Alexander.Mehler@uni-bielefeld.de), soweit diese noch vorhanden sind, da die einzige Alternative darin besteht, die betroffenen Beiträge zu scannen.

Bielefeld und Regensburg, im Dezember 2005

Alexander Mehler und Christian Wolff

Inhaltsverzeichnis

LDV-FORUM – Band 20(2) – 2005
Themenheft Korpuslinguistik

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

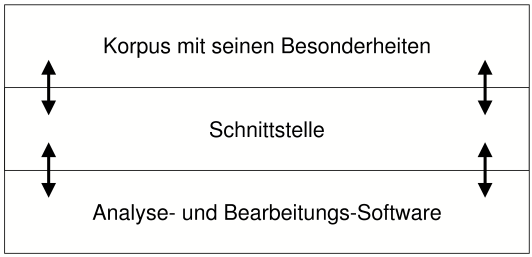


Abbildung 1: Das Prinzip der Korpus-Schnittstelle.

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.

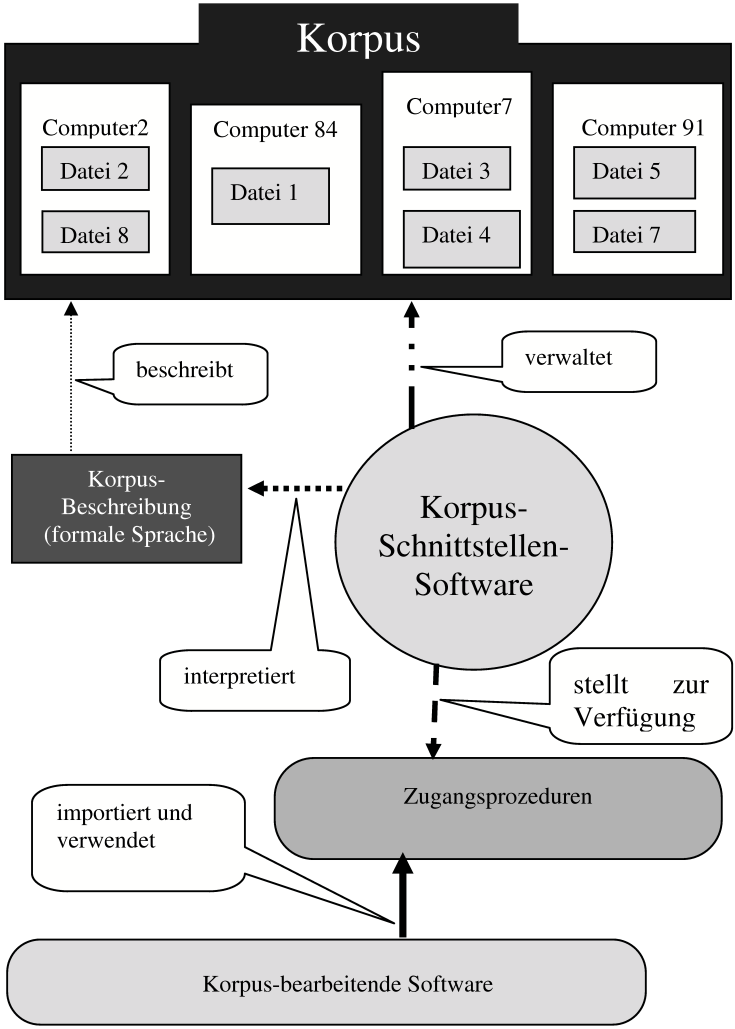


Abbildung 2: Eine Architektur einer allgemeinen Korpus-Schnittstelle.

Thorsten Trippel, Thierry Declerck und Ulrich Heid

Sprachressourcen in der Standardisierung

Wir berichten über internationale Normungsarbeit im Bereich von Sprachressourcen. Die Normen werden von internationalen Arbeitsgruppen im Rahmen der *International Organization for Standardization* (ISO) entwickelt und jeweils national von entsprechenden Gruppen, in Deutschland koordiniert vom Deutschen Institut für Normung (DIN), begleitet und diskutiert. Für die automatische Sprachverarbeitung besteht seit Jahren zunehmend Bedarf an elektronischen Ressourcen: Lexika, Korpora, Grammatiken, Annotationskonventionen, Sprachdatensammlungen, usw. Damit solche Ressourcen über einen einzelnen Anwendungskontext hinaus wiederverwertbar sind und zwischen Arbeitsgruppen ausgetauscht werden können, wird an einer Normung ihrer Repräsentationsformate und der zur Beschreibung von Ressourceninhalten benutzbaren Vokabularien gearbeitet (Datenkategorien). Waren in der Vergangenheit Standardisierungsbemühungen auf bestimmte Ausschnitte aus dem Spektrum der linguistischen Beschreibungen von Ressourcen beschränkt (z.B. die EU-Projekte SAM im Bereich gesprochener Sprache, EAGLES und ISLE im Bereich von Morphosyntax, Syntax, lexikalischer Semantik in Texten und Lexika und Sprachtechnologie), so ist die Zielsetzung der 2002 und 2003 gegründeten ISO (TC 37 SC 4) bzw. DIN (NAT AA 6) Arbeitsgruppen breiter: es geht um Metarichtlinien für die Repräsentation und Annotation von Texten ebenso wie um Datenkategorien für Lexika, morphologische und morphosyntaktische Analyse, usw. Wir beschreiben den aktuellen Stand der Normungsdiskussion.

1 Einführung in die Normierung für Sprachressourcen: Historischer Kontext

Sprachressourcen sind eine Klasse heterogener Informationen, die Gegenstand von Linguistik und Sprachtechnologie sind, aber auch in Anwendungskontexten wie Übersetzung und Lexikonentwicklung gefragt sind. Dazu gehören Textkorpora, Lexika, Daten gesprochener Sprache, aber auch Annotationsrichtlinien und -verfahren. Die verschiedenen Ressourcen unterscheiden sich dabei häufig sowohl durch ihre Form, die verwendeten Datenstrukturen und Anwendungskontexte. Ebenso variieren die Annotationsstrukturen sehr oft von Projekt zu Projekt oder zwischen verschiedenen Anbietern von Ressourcen. In der Praxis läuft dies in der Regel auf idiosynkratische Bestimmungen von Datenformaten und Verarbeitungsverfahren hinaus; Konsistenzprüfungen werden, wenn überhaupt, ad hoc von Experten oder von applikationsspezifischen Parsern durchgeführt. Diese idiosynkratischen Strukturen erlauben es aber nicht, Daten zwischen Applikationen oder Nutzern auszutauschen, ohne vorher eine detaillierte Analyse und Transformation

durchzuführen, selbst wenn Teilstrukturen und Informationsgehalt direkt vergleichbar wären.

Bei vielen Verfahren, insbesondere für statistische Ansätze, besteht ein erheblicher und stetig größer werdender Bedarf an hochwertigen Ressourcen. Daher wird eine Standardisierung von Formaten und Formalismen über Grenzen der linguistischen Theoriebildung hinweg angestrebt, um zumindest existierende Werkzeuge und Verfahren verwenden und austauschen zu können, aber auch um Ressourcen selbst als Grundlage auch anderer als der ursprünglich intendierten Verwendungen benutzen zu können.

Der vorliegende Beitrag beruht auf den Normentwürfen und Vorschlägen aus dem Standardisierungsgremium der International Organization for Standardization (ISO) im Rahmen des technischen Komitees 37, Arbeitsausschuss 4 (*Language Resources*) ISO TC 37/SC 4. Das Komitee wurde im Sommer 2002 gegründet und ist international besetzt; dabei entsenden interessierte nationale Normungsinstitutionen Experten, die an Normentwürfen mitarbeiten. In Deutschland, wie in anderen Ländern, gibt es seit Mai 2003 eine nationale Arbeitsgruppe, die deutsche Beiträge zur internationalen Normung koordiniert. Sie wird organisatorisch vom Deutschen Institut für Normung (DIN) im Rahmen seines Normungsausschusses Terminologie betreut. Die Autoren sind Mitglieder dieser Arbeitsgruppe.

Die Arbeitsgruppe ISO TC37/SC4 führt auf internationaler Ebene Versuche der Normierung fort, die in den 1990er Jahren in einzelnen Projekten der Sprachverarbeitung begonnen worden waren. Beispiele solcher früheren Versuche für Vereinheitlichung von Annotationen und Annotationsverfahren sind etwa die EU-Projekte SAM für die Sprachsignalannotation, EAGLES (Expert Advisory Groups on Linguistic Engineering Standards) für morphosyntaktische und syntaktische Annotation von Textkorpora (URL: `http://www.ilc.cnr.it/EAGLES96/annotate/annotate.html` und `http://www.ilc.cnr.it/EAGLES96/segsasg1/segsasg1.html`) oder ISLE (International Standards for Language Engineering), z.B. zur Repräsentation von Wörterbucheinträgen in NLP-Systemen (vgl. MILE). Diese Vorhaben waren auf eine Harmonisierung bzw. Standardisierung der Annotationen selbst ausgerichtet; die Harmonisierung sollte durch einen Konsensus-Prozeß erreicht werden: eine Art minimaler, allgemein akzeptierter Basisvorschlag für die jeweilige Annotation wurde erarbeitet.

Demgegenüber ist es das Kennzeichen eines Teils der hier diskutierten Normungsvorhaben, dass die Standardisierung *eine Ebene höher* ansetzt, auf der Schicht von Meta-Annotationen, von Frameworks für die Erstellung und den Austausch von Annotationen, Datenstrukturen und Ressourcen, oder bei Prozeduren für die Erstellung von Inventaren für Datenkategorien. Ein Teil der Normung ist also nicht mehr auf Harmonisierung der Ressourcen durch gemeinsame Formate, sondern auf Interoperabilität durch gemeinsame Meta-Formate, Austauschformate, Herangehensweisen usw. gerichtet. Jeder, der Ressourcen produziert, soll seine Daten in ein solches Format abbilden können; jeder der fremde Ressourcen nutzt, soll die Gewähr haben, eine interpretierbare *Übersetzung* oder Transformation leisten zu können.

2 Einsatzbereite Standards und Standardentwürfe

Im Rahmen der ISO Standardisierung gibt es verschiedene Phasen, bevor eine Norm als verbindlich anzusehen ist, nämlich *Work Item*, *Committee Draft* (CD), *Draft International Standard* (DIS), *Final Draft International Standard* (FDIS). Das Work Item ist dabei nur eine Beschreibung eines Normierungsvorhabens, CD der erste in den Normenausschüssen zur Diskussion stehende Entwurf einer Norm, DIS ist eine entsprechend fortgeschrittene Version, die auch interessierten Kreisen außerhalb der Standardisierungsgremien zugänglich gemacht werden kann, und FDIS ist eine fast endgültige Version, die sich schon zur Implementierung in Testumgebungen eignet.

Derzeit sind im Bereich der Sprachressourcen einige Standards in der Entwicklung; diese betreffen die folgenden Fragestellungen:

allgemeine (linguistische) Annotation: Grundlagen für die Kodierung linguistischer Informationen

Wörterbuchbeschreibungen: Beschreibung und Austausch von Wörterbucheinträgen und ganzen Wörterbüchern

Wortsegmentierung: sprachübergreifende Kriterien zur Beschreibung von Wortgrenzen

morphosyntaktische Annotation: einheitliche Annotation von morphosyntaktischer Information

syntaktischen Annotation: einheitliche Annotation von syntaktischer Information, ein neues Work Item

Merkmalsstrukturen: Kodierung von Merkmalsstrukturen verschiedener linguistischer Theorien

Datenkategorien: Definitionen und Beschreibung der Relation verschiedener linguistischer Datenkategorien.

Diese Normen werden im Folgenden kurz charakterisiert und diskutiert. Die Standards, die heute schon einsetzbar sind, werden durch Beispiele exemplifiziert.

2.1 Grundlagen linguistischer Annotation: Linguistic Annotation Framework (LAF)

Durch die Verwendung existierender Konventionen aus dem World Wide Web Consortium (W3C) wie XML (Bray et al., 2004), RDF (Beckett, 2004), OWL (McGuinness und van Harmelen, 2004), etc. versucht das Linguistic Annotation Framework eine einheitliche Grundlage für die Annotation von linguistischen Daten zu legen. Dabei liegt ein Schwerpunkt auf höheren Annotationsebenen, etwa morphosyntaktische, syntaktische und semantische Annotation, die auf tieferen Ebenen aufsetzen, ohne dabei gegenüber anderen Bereichen abgeschlossen zu sein.

Die auf der Grundlage verschiedener Bedürfnisse entwickelten Annotationsstandards, z.B. die Ergebnisse von EAGLES (Calzolari und McNaught, 1996; Leech und Wilson, 1996), ISLE (Atkins et al., 2002, 2003), etc. zu Morphosyntax, Syntax, Semantik und Lexikon haben zu einer Vielzahl von Inkompatibilitäten geführt. Um eine gemeinsame Basis existierender Annotation zu finden, wird daher mit Hilfe einer allgemeinen Merkmalsstruktur auf Grundlage von Datenkategorien, die ebenfalls zu standardisieren sind, ein generisches Datenformat definiert. Bestehende Annotationen sind daher in dieses Format transformierbar. Ziel ist also ein Metaformat, das es erlauben soll, linguistische Annotationen auszudrücken und auszutauschen.

Dabei stellt die Definition der Datenkategorien genauso ein Problem dar wie die Verwendung verschiedener Merkmalsstrukturhierarchien, die auf unterschiedlichen theoretischen Annahmen herrühren können. Das Problem der Definition von Datenkategorien soll dabei durch ein offenes Datenkategorien-Repository gelöst werden (siehe Abschnitt 2.7), wodurch eine maximale Unabhängigkeit von spezifischen Theorien möglich wird. Interessierte Kreise sollen die Möglichkeit erhalten, Vorschläge für Datenkategorien zu machen. Alle von einem dafür benannten Gremium akzeptierten Datenkategorien werden samt Beschreibung und Beispielen zentral gesammelt und jedem Benutzer zur Verfügung gestellt. Die Merkmalsstrukturhierarchie ist dagegen nicht als linguistische Theorie per se zu betrachten, auch wenn sie unter Umständen eine bestimmte linguistische Theorie abbildet, sondern nur als Austauschformat. Ob diese Trennung zwischen Theorie und Austauschformat allerdings vollständig erreicht werden kann, ist noch nicht beschrieben worden.

2.2 Grundlagen lexikalischen Markups: Lexical Markup Framework (LMF)

Im Bereich der Terminologiedatenbanken gibt es die Bestrebungen, für den Austausch ein Framework zu definieren, das als Grundlage für die Überarbeitung des *Machine Readable Terminology Interchange Format* (MARTIF, ISO 12200) dienen soll. Analog dazu werden derzeit Standards für die Beschreibung lexikalischer Datenbanken, insbesondere Wörterbücher mit definitorischen Inhalten, entwickelt, in denen formale Oberflächenmerkmale und Semantik voneinander getrennt strukturiert werden, und in denen lexikalische Strukturen eindeutig modelliert werden.

Verschiedene Applikationen, die Lexika verwenden, legen unterschiedliche Datenmodelle zugrunde. Um zu einer Vereinheitlichung dieser Datenmodelle für semantische Lexika zu kommen, beziehen sich die in der Entwicklung befindlichen Normen auf das Lexical Markup Framework (LMF), das eine Repräsentation lexikalischer Information in einem einheitlichen Modell darstellt, durch welches zunächst zumindest die Inhalte allgemeiner einsprachiger Definitionswörterbücher und ähnlich strukturierter Lexikondatenbanken repräsentiert werden können.

Derzeit ist bei der Erstellung des *LMF* nicht beschrieben, wo Grenzen liegen, wodurch nicht klar ist, welche lexikalischen Ressourcen mit seiner Hilfe abgebildet werden können. Da dieser Standard aber analog zu terminologischen Ressourcen im Terminology Markup Framework (Neufassung von ISO 12200:1999 (1999)) definiert wird, ist an die

Beschreibung semantischer Lexika im Sinne der lexikalischen Semantik zu denken. Eine Behandlung der Grenzen sollte jedoch Gegenstand von weiteren Normentwürfen sein, bevor dieser Standard verabschiedet werden kann.

2.3 Die Segmentierung von geschriebenen Wörtern für die Informationsverarbeitung

Als Grundlage für die automatische Verarbeitung von Sprachressourcen wird in der Regel davon ausgegangen, dass man verschiedenen Wörter im elektronisch verfügbaren Text voneinander unterscheiden kann. Dies setzt voraus, dass man eine Möglichkeit hat, Grenzen zwischen Wörtern zu ziehen. In westlichen Sprachen mit lateinischer Schrift wird diese Segmentierung von Wörtern durch Leerzeichen als typographische Konvention deutlich gemacht, was allerdings nicht für andere Schriftsysteme gelten muss, und in der Tat nicht allgemeingültig ist. Die Schreibung von Chinesisch, Japanisch und Koreanisch kommt z.B. traditionell ohne Leerstellen zwischen Wörtern aus (der aktuelle Normungsvorschlag ist übrigens von chinesischen Wissenschaftlern als *New Work Item* auf den Weg gebracht worden). Die Differenzierung von Wörtern und Phrasen aufgrund von linguistischen Kriterien ist auch nicht so allgemein, wie es Programmiersprachen antizipieren, die Wortgrenzen bei Interpunktionszeichen oder Leerzeichen als erreicht ansehen. Ein Beispiel dafür ist die nicht notwendigerweise eindeutige Segmentierung von Komposita, etwa durch verschiedene Möglichkeiten der Zusammen- oder Getrenntschreibung, die durch die Schreibung mit Binde-Strich (wie in diesem Wort) auch ein Hybrid kennen. In einem Handbuch eines deutschen Industrieunternehmens finden sich beispielsweise eine Vielzahl von Varianten für Begriffe wie *Gasmeßgerät*, *Gas-Meßgerät* und *Gasmeß-Gerät* nebeneinander. Ähnliches gilt z.B. in Paralleltexten, wenn etwa im Deutschen Komposita zusammen geschrieben werden, werden in relativ nah verwandten Sprachen wie Englisch dagegen als Kombination von zwei oder mehr typographischen Wörtern dargestellt.

Ziel des Normungsvorhabens ist eine Vereinheitlichung, um z.B. Benchmarks zu Evaluationszwecken von sprachverarbeitenden Systemen definieren zu können, primär mit dem Ziel einer Wortdefinition für nicht lateinische Schriftsysteme ohne Wortgrenzenkonventionen. Mittelfristig können die Ergebnisse von Wortsegmentierungsverfahren in der Informationsrecherche und der Terminologieextraktion unmittelbar eingesetzt werden.

Basierend auf linguistischen Regeln, Häufigkeit und Stabilität von Zeichenkombinationen wird die *Wortheit* von Mehrwort-Ausdrücken auf der Grundlage von Wortlisten aus Korpora bestimmt, und es wird ein Metamodell für die Segmentierung von Wörtern definiert.

Ein wesentliches Problem besteht in den unterschiedlichen Auffassungen zu Wortgrenzen und in der Verwendung von Worteinheiten in existierenden Systemen. Dieses Problem ist etwa aus der Praxis des Übersetzungswesens hinreichend bekannt, in dem Übersetzungsumfänge nach Wortzahl bewertet werden. Dies führt bereits bei der Bewertung von agglutinierenden Sprachen zu Problemen. Ein einheitliches Vorgehen wäre also auf Grund von praktischen Erwägungen sinnvoll, um eine einheitliche Bezugsgröße

definieren zu können. Das Vorhaben ist Ende 2005 auf dem Stand eines zur Normierung vorgeschlagenen Work Items.

2.4 Grundlagen für Morphosyntaktische Annotation: Morphosyntactic Annotation Framework (MAF)

Das Ziel des *Morphosyntactic Annotation Framework* ist eine einheitliche Kodierung von morphosyntaktischen Informationen, die in Datenströmen enthalten sind, also sowohl im Bereich der textuellen Daten als auch zur Signalannotation.

Der Entwurf des Morphosyntaktischen Annotations Frameworks (MAF) besteht aus zwei Teilen:

1. Die Segmentierung, also die Bestimmung der Wörter, die Behandlung von Ambiguitäten und die formale Beschreibung von internen Strukturen mit Hilfe von Merkmalsstrukturen (Attribut-Werte Paaren)

2. Eine inhaltliche Beschreibung der morphosyntaktischen Annotation, also eine Angabe zur Einbettung strukturierter Informationen. Dies schließt auch die Möglichkeit der multiplen Annotation mit ein, etwa für Numerus, Genus, Tempus, etc., weil ja viele Formen synkretistisch sind (*Hunde:* `nom/gen/acc plural`).

Diese Norm bezieht sich dabei unmittelbar auf relevante Datenkategorien zur Beschreibung der morphosyntaktischen Annotation. Ferner gibt es auch Anknüpfungen zur Segmentierungproblematik, da zu klären wäre, wie etwa für Deutsch Komposita zu behandeln sind, als mehrere Wörter oder als lexikalische Einheit. Die Arbeiten sind Ende 2005 bis zu einem Committee Draft gediehen.

2.5 Syntactic Annotation Framework (SynAF)

Innerhalb des eContent Projekts *LIRICS* (siehe auch `http;//lirics.loria.fr` und Sektion 3 unten), wird an einem Normvorhaben für syntaktische Annotationen gearbeitet. Ein entsprechendes Work Item wurde dazu dem ISO Commitee TC 37/SC4 vorgelegt und bereits akzeptiert.

Das *Syntactic Annotation Framework* (SynAF) verfolgt primär zwei Ziele:

1. Die Definition eines Metamodells für syntaktische Annotationen (ähnlich wie für die Segmentierung oder die Morphosyntax, wie weiter oben beschrieben).

2. Die Aufstellung einer Liste von Datenkategorien (s. Ide und Romary (2004)) als Grundlage für eine einheitliche syntaktische Annotation.

Die Standardisierungsarbeit von SynAF basiert auf den neuesten Entwicklungen im Bereich der syntaktischen Annotation, sowohl als Ausgabe von Parsern, die oft dem Zweck der Theorievalidierung dienen, als auch als bestehende Baumbanken, die primär als Trainingsdaten für Analysesysteme dienen. Das Metamodell und die Datenkategorien, die in

SynAF definiert werden, sollen dann die Interoperabilität und die Wiederverwendbarkeit von diesen zwei Typen von Ressourcen unterstützen.

Das Metamodell von SynAF muss flexibel genug sein, um die zwei Haupttypen von syntaktischen Annotationen abzudecken: Konstituentenstrukturen und Dependenzstrukturen.

Als Eingabematerial für SynAF werden folgende Ressourcen verwendet:

- Zum einen so genannte *legacy data*, die in Baumbanken zu finden sind, wie zum Beispiel innerhalb der *Penn Treebank*.
- Zum anderen bestehende Grammatiken, welche die syntaktischen Strukturen für verschiedene Sprachen abdecken.

Die Ausgangsbasis für SynAF besteht demnach in Korpora, die syntaktische Konstituenz und Dependenz kombinieren, wie TIGER (Uszkoreit, 2003) für das Deutsche, oder ISST (Montemagni et al., 2002) für das Italienische, aber auch Korpora zu nicht-europäischen Sprachen (siehe auch Abeillé et al. (2003)). Ebenfalls werden Ausgaben von syntaktischen Parsern berücksichtigt, die in verschiedenen Kontexten und Anwendungen entwickelt wurden (zum Beispiel HPSG, Pollard und Sag (1987), LS-GRAM (siehe auch LS-GRAM (2005) in der Bibliographie) und LFG Grammatiken (siehe hierzu die Referenz zum LFG Pargram Projekt (2005)) oder flache und robuste Grammatiken).

SynAF wird sich auch dem Thema der syntaktischen Ambiguitäten widmen (aufbauend hier auf bereits existierenden Vorschlägen, die in MAF gemacht worden sind, siehe Clément und de la Clergerie (2005)). Auch das Thema der mehrschichtigen Annotationen wird von SynAF angesprochen (zum Beispiel für die parallele Beschreibung von flachen vs. tiefen Analysen). Hier wird SynAF sich in das Linguistic Annotation Framework (LAF) einfügen. Dies spielt für die Beschreibung sogenannter langer Abhängigkeiten eine Rolle, die häufig eine eigene Annotationsschicht (*layer*) brauchen, um repräsentiert zu werden.

Ferner wird diskutiert, ob SynAF auch Information über syntaktischen Operationen beschreiben können soll, d.h. ob die Annotation auch Angaben über die beteiligten Prozesse auf dem Weg zum Analyseergebnis aufnehmen soll.

2.6 Repräsentation von Merkmalsstrukturen: Feature Structure Representation

Merkmalsstrukturen sind übliche Formalismen zur Beschreibung von Strukturen in vielen linguistischen Theorien und Ansätzen, etwa in der HPSG, LFG, im generativen Lexikon, etc, wo hierarchisierte Merkmalsstrukturen Verwendung finden. Abbildung 1 zeigt so eine Merkmalsstruktur für das englische Wort *dog* im HPSG-Paradigma.

Merkmalsstrukturen weisen einen hohen Informationsgehalt auf, sind sehr stark formalisiert und bieten sich daher für die automatisierte Verarbeitung an. Für die Verarbeitung im Bereich automatisierter Systeme werden dafür komplexe Strukturen und Programme eingesetzt, wobei nicht zuletzt die Komplexität den Bedarf nach einem einheitlichen Formalismus für den Austausch von Merkmalsstrukturen über einzelne

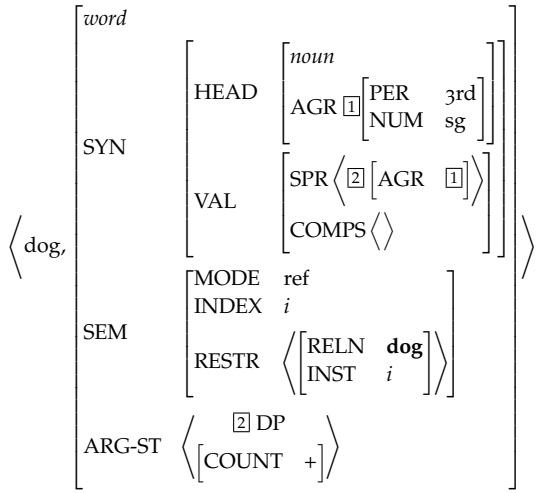


Abbildung 1: HPSG-Lexikoneintrag in Merkmalsstrukturform für das englische Wort *dog*, aus Sag et al. (2003), S. 254.

Systeme hinaus begründet. Hierzu kann man auf Vorarbeiten der Text Encoding Initiative (TEI, siehe Sperberg-McQueen und Burnard (2004)) zurückgreifen, die eine Syntax für die Beschreibung von Merkmalsstrukturen entwickelt hat.

Eine mögliche Repräsentation der in Abbildung 1 angegebenen Struktur gemäß einem Normentwurf ISO CD 24610-1:2003 (2003) stellt der Ausschnitt aus einem XML-Baum in Tabelle (1 – am Ende dieses Beitrags) dar.

Bei der Beschreibung dieser Merkmalsstrukturen gibt es natürlich Anknüpfungspunkte an die Problematiken anderer Standardentwürfe. So ist mit der Festlegung der Beschreibung von Merkmalsstrukturen zunächst einmal nicht festgelegt, inwieweit linguistische Theorien von ihren jeweiligen Repräsentationsformalismen, in diesem Fall also den getypten Merkmalsstrukturen, abzukoppeln sind. Das Problem der Trennung zwischen Repräsentationsformat und Theorie wird in dem Moment offensichtlicher, in dem Datenkategorien verschiedener linguistischer Theorien, entweder mit unterschiedlichem Namen und unterschiedlicher Semantik, mit unterschiedlichem Namen aber gleicher Bedeutung, oder gar mit gleichem Namen aber unterschiedlicher Bedeutung auftreten.

Zur Validierung der Angemessenheit des als Norm vorgeschlagenen Beschreibungsverfahrens sollten daher verschiedene linguistische Analysesysteme als Referenz implementiert und auf Interoperabilität zwischen den Merkmalshierarchien untersucht werden. Die Benennung von Datenkategorien ist dabei Gegenstand eigenständiger Normierungsbestrebungen. Nichtsdestotrotz ist die Beschreibung von Merkmalsstrukturen weit fortgeschritten und die Verabschiedung als internationaler Standard ISO 24610-1 darf für die nähere Zukunft erwartet werden.

2.7 Datenkategorien für elektronische lexikalische Ressourcen: Terminology and other language resources – Data categories

In der Sektion über die Merkmalsstrukturen wurde kurz auf das Problem der Bedeutung-Namens-Paaren in verschiedenen linguistischen Theorien hingewiesen. Ziel bei der Standardisierung von Datenkategorien für Sprachressourcen ist es, Datenkategorien für die Verwendung in lexikalischen Datenbanken zu definieren. Dabei wird von einer korrespondierenden, in Überarbeitung befindlichen Norm für Datenkategorien aus dem Bereich der Terminologie ausgegangen (ISO 12620:1999, 1999). Da erste Studien gezeigt haben, dass es erhebliche Überschneidungen zwischen Datenkategorien in Terminologie und Lexikographie gibt, wurde ein Vorschlag zur Trennung in verschiedene Standards mit einer allgemeinen Methodenspezifikation und separaten Kategoriebeschreibungen für Terminologie und Lexikographie verworfen.

Ausgehend von zentralen Datenkategorien werden bei der Standardisierung von Datenkategorien Mechanismen normiert, die der Erweiterung der Datenkategoriebasis dienen sollen (siehe auch Ide und Romary (2004)). Dies soll gewährleisten, dass Datenkategorien interoperabel sind und auch bei neueren oder verbesserten Theorien standardisierte Datenkategorien Verwendung finden können (einschließlich *Rückwärtskompatibilität*). Daher wurde zwar eine lange Liste von Datenkategorien erstellt, die teilweise mit Subkategorien versehen sind, zusammen mit einer Definition der Kategorie, aber der Schwerpunkt liegt auf der Definition eines Datenkategorie-Registers.

Eine besondere Komplexität erhält dieses *Data Category Repository* dadurch, dass es Offenheit gegenüber neuen Entwicklungen verlangt, was dazu führt, dass man Mechanismen zur Aufnahme von neuen Datenkategorien definieren muss, die ebenfalls unabhängig von Vorlieben und theoretischen Annahmen sind. Allerdings muss sichergestellt werden, dass die Offenheit nicht dazu führt, dass äquivalente Datenkategorien unabhängig voneinander definiert werden, was dem Grundsatz der Austauschbarkeit diametral entgegenstünde.

Diese Diskussion zeigt, dass die Normierung auf der einen Seite weit fortgeschritten ist, indem bereits ein Grundgerüst an Datenkategorien existiert, aber die Erweiterungsfunktionalität des Datenkategorieregisters und die eindeutige Beschreibung der Modalitäten, wie es ergänzt werden soll, sich noch in der Entwicklung befindet.

3 Einsatz von Standards in der Praxis

In vielen Feldern ist unmittelbar klar, dass es einen Bedarf an Standards für linguistische Annotationen gibt. Ein Beispiel dafür ist die transferbasierte maschinelle Übersetzung: Wenn standardisierte syntaktische Annotationen für Quell- und Zielsprache vorliegen, ist zu erwarten, dass Übersetzungssysteme mit geringerem Aufwand auf Seiten der Trainingsdaten erstellt werden können.

Die Anwendungen gehen jedoch weit über die Sprachverarbeitung hinaus, insbesondere in den Bereich des *Semantic Web* (Berners-Lee et al., 2001), einer Erweiterung des World Wide Webs. Damit das Semantic Web tatsächlich funktionieren kann, müssen Webseiten

semantisch annotiert werden. Im Bereich des *Semantic Web* versteht man unter semantischer Annotation dabei eine Verarbeitung, die einen Text mit Informationen anreichert, die aus Wissensbasen stammen, also aus Datenbanken, Taxonomien, Ontologien, etc. Es gibt aber wenige Werkzeuge, die diese Arbeit unterstützen, und selbst die existierenden Werkzeuge können nicht darüber hinweg täuschen, dass die semantische Auszeichnung extrem zeitintensiv ist. Daher gibt es Bemühungen, diese Art der Annotation zu automatisieren, und zwar auf der Grundlage von sprachverarbeitenden Werkzeugen, die den Webdokumenten eine (linguistische) syntaktische Struktur verleihen, bevor sie dann auf die Wissensbasen abgebildet werden, um semantische Annotationen zu generieren. Standardisierte linguistische Annotationen würden diesen Abbildungsprozess erheblich erleichtern (Buitelaar und Declerck, 2003).

Im gleichen Kontext wird nach Möglichkeiten gesucht, Wissensbasen automatisch aus größeren Dokumentmengen zu extrahieren, wobei auch maschinelles Lernen eingesetzt wird. Dieses Verfahren verlangt eine größere Menge von linguistischen Annotationen, die speziell auch Dependenz-Relationen aufweisen, damit sogenannte RDF-Tripel erzeugt werden können. Diese RDF-Tripel kann man sich als Subjekt, Objekt und Prädikat vorstellen, d.h. einem Gegenstand wird mittels eines Verbs eine Eigenschaft zugewiesen.

Um verfügbare linguistische Annotationen z.B. in Form von Baumbanken oder Ergebnissen von Parsern für Werkzeuge im Semantic-Web-Kontext verfügbar zu machen, müssen diese Ressourcen auf standardisierte Annotationen abgebildet werden, da es wesentlich einfacher ist aus einer großen Menge von standardisierten Annotationen Wissen zu akquirieren, als aus heterogenen oder gar idiosynkratisch annotierten Dokumenten.

Die Mitarbeit an Normungsaktivitäten und die Entwicklung von Standards basiert auf den fachlichen Interessen und Bedürfnis in verschiedenen Projekten, Vorhaben und Unternehmen. Auf Initiative einer französischen Forschungsorganisation (LORIA) wurde zusätzlich ein europäisches Projekt im Rahmen des *eContent*-Programms eingereicht und bewilligt, das sich schwerpunktmäßig auf die Erforschung von Ressourcen, sowie auf die benötigte standardisierte Infrastruktur konzentriert.

Das Projekt *LIRICS* (Linguistic Infrastructure for Interoperable Resources and Systems) läuft seit dem 1. Januar 2005 und treibt auf europäischer Ebene einige der oben vorgestellten Themen in enger Kooperation mit der ISO voran. So wurde zum Beispiel das ISO-Vorhaben *SynAF* (vgl. Abschnitt 2.5 oben) innerhalb von *LIRICS* initiiert. *LIRICS* entwickelt auch eine open-source Implementierung, Webservices und Testsuites für neun Sprachen, welche die Implementierung einiger der oben besprochenen Standards unterstützen und validieren. Die Notwendigkeit der Standardisierung von Sprachressourcen und ihre wirtschaftliche Relevanz für die Generierung von digitalen Inhalten ist dadurch auch auf europäischer Ebene dokumentiert und wird aktiv unterstützt.

Literatur

Abeillé, A., S. Hansen-Schirra und H. Uszkoreit (Hrsg.) (2003). *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, Budapest.

Atkins, S., N. Bel, F. Bertagna, P. Bouillon, N. Calzolari, C. Fellbaum, R. Grishman, A. Lenci, C. MacLeod, M. Palmer, G. Thurmair, M. Villegas und A. Zampolli (2002). From resources to applications. Designing the multilingual ISLE lexical entry. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas de Gran Canaria, S. 687–693.

Atkins, S., N. Bel, P. Bouillon, T. Charoenporn, D. Gibbon, R. Grishman, C.-R. Huang, A. Kawtrakul, N. Ide, H.-Y. Lee, P. J. K. Li, J. McNaught, J. Odijk, M. Palmer, V. Quochi, R. Reeves, D. M. Sharma, V. Sornlertlamvanich, T. Tokunaga, G. Thurmair, M. Villegas, A. Zampolli und E. Zeiton (2003). Standards and best practice for multilingual computational lexicons and MILE (the multilingual ISLE lexical entry). Deliverable D2.2-D3.2 ISLE computational lexicon working group, International Standards for Language Engineering (ISLE), Pisa. Entstehungsjahr anhand von Dateimetadaten verifiziert.

Beckett, D. (2004). RDF/XML syntax specification (revised). URL: `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`.

Berners-Lee, T., J. Hendler und O. Lassila (2001). The Semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*.

Bray, T., J. Paoli, C. M. Sperberg-McQueen, E. Maler und F. Yergeau (2004). Extensible Markup Language (XML) 1.0 (third edition). URL: `http://www.w3.org/TR/2004/REC-xml-20040204/`.

Buitelaar, P. und T. Declerck (2003). Linguistic annotation for the Semantic Web. In S. Handschuh und S. Staab (Hrsg.), *Annotation for the Semantic Web*. Amsterdam: IOS Press.

Calzolari, N. und J. McNaught (1996). Synopsis and comparison of morphosyntactic phenomena encoded in lexicons and corpora: A common proposal and applications to european languages. Technical report, Expert Advisory Group on Language Engineering Standards (EAGLES).

Clément, L. und É. de la Clergerie (2005). MAF: A morphosyntactic annotation framework. In *Proceedings of the 2nd Language and Technology Conference (LT'05)*, Poznan, S. 90–94.

Ide, N. und L. Romary (2004). A registry of standard data categories for linguistic annotation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon.

ISO 12200:1999 (1999). Computer applications in terminology – machine-readable terminology interchange format (MARTIF) – negotiated interchange. Technical report, ISO.

ISO 12620:1999 (1999). Computer applications in terminology – data categories. Technical report, ISO.

ISO CD 24610-1:2003 (2003). Language resource management – feature structures – part 1: Feature structure representation. Technical report, ISO. Committee Draft.

Leech, G. und A. Wilson (1996). Recommendations for the morphosyntactic annotation of corpora. Technical report, Expert Advisory Group on Language Engineering Standards (EAGLES).

LFG Pargram Projekt (2005). LFG parallel grammar project. URL: `http://www2.parc.com/istl/groups/nltt/pargram/` – Stand: 6. Dezember 2005.

LS-GRAM (2005). LS-GRAM project. URL: `http://www.iai.uni-sb.de/iaide/en/ls_gram.htm` – Stand: 6. Dezember 2005.

McGuinness, D. L. und F. van Harmelen (2004). OWL web ontology language overview. URL: http://www.w3.org/TR/2004/REC-owl-features-20040210/.

Montemagni, S., F. Barsotti, M. Battista, N. Calzolari, A. Lenci, O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Basili, R. Raffaelli, M. Pazienza, D. Saracino, F. Zanzotto, F. Pianesi, N. Mana und R. Delmonte (2002). Building the Italian syntactic-semantic treebank. In A. Abeillé (Hrsg.), *Building and Using syntactically annotated corpora*, S. 189–210. Dordrecht: Kluwer.

Pollard, C. und I. Sag (1987). *Head-Driven Phrase Structure Grammar*. CSLI and University of Chicago Press.

Sag, I. A., T. Wasow und E. M. Bender (2003). *Syntactic Theory* (2. Aufl.). Stanford: CSLI Publications.

Sperberg-McQueen, C. M. und L. Burnard (2004). TEI P4 guidelines for electronic text encoding and interchange XML-compatible edition. URL: http://www.tei-c.org/P4X/.

Uszkoreit, H. (2003). TIGER project. URL: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/ – Stand: 6. Dezember 2005.

```
<fs>
  <f org="list" name="dog">
     <fs>
       <f name="orth"><str>dog</str></f>
       <f name="word"/>
       <f name="syn">
         <fs>
             <f name="head">
                <fs>
                   <f name="noun"/>
                   <f name="agr">
                      <fs id="one">
                          <f name="per"><str>3rd</str></f>
                          <f name="num"><str>sg</str></f>
                      </fs>
                   </f>
                </fs>
             </f>
              <f name="val">
                <fs>
                   <f name="spr" org="list">
                      <fs>
                         <f name="null" fVal="two"/>
                         <f name="agr" fVal="one"/>
                      </fs>
                   </f>
                   <f name="comps" org="list"/>
                </fs>
              </f>
         </fs>
       </f>
       <f name="sem">
          <fs>
              <f name="mode"><str>ref</str></f>
              <f name="index"><str>i</str></f>
              <f name="restr">
                <fs>
                    <f name="reln"><str>dog</str></f>
                    <f name="inst"><str>i</str></f>
                </fs>
              </f>
         </fs>
       </f>
       <f name="arg-st" org="list">
         <fs id="two"><f name="dp"/></fs>
         <fs>
              <f name="count"><plus/></f>
          </fs>
       </f>
     </fs>
  </f>
</fs>
```

Tabelle 1: Eine mögliche XML-basierte Repräsentation der in Abbildung 1 angegebenen Struktur.

Emmerich Kelih, Peter Grzybek

Satzlänge: Definitionen, Häufigkeiten, Modelle (Am Beispiel slowenischer Prosatexte)

1 Einleitung

Die vorliegende Untersuchung versteht sich als ein Beitrag zur Satzlängenforschung. Nach einleitender Darstellung der Analysemöglichkeiten auf der Ebene der Satzlängen, geht es hauptsächlich um die Diskussion der Anwendung von unterschiedlichen Satzdefinitionen. Auf der Basis eines Korpus slowenischer Texte wird der Frage nachgegangen, welchen Einfluss die Anwendung unterschiedlicher (durchaus üblicher) Satzdefinitionen auf (a) deskriptive Kenngrößen der Häufigkeitsverteilung hat, und (b) inwiefern davon die Adäquatheit und Güte theoretischer Verteilungsmodelle abhängt.

2 Satzlänge: Stilcharakteristikum – Textklassifikation – Modellierung

Untersuchungen zu Satzlängen werden mit den unterschiedlichsten Fragestellungen im Rahmen eines breiten Forschungsspektrums durchgeführt: Zum Einen wird die Satzlänge in erster Linie als ein (1) spezifisches Stilcharakteristikum betrachtet; zum Anderen wird die Häufigkeitsverteilung von Satzlängen vor allem darauf hin untersucht, inwiefern sich diese durch (2) theoretische Wahrscheinlichkeitsmodelle beschreiben lässt. Beide Bereiche[1] sind im folgenden einleitend etwas ausführlicher zu kommentieren.

Ad 1. Im Bereich von stilistischen Untersuchungen ist die Bedeutung der Satzlänge früh erkannt worden (vgl. Sherman, 1888). Ohne an dieser Stelle einen erschöpfenden Überblick geben zu wollen, lässt sich zeigen, dass sich in der aktuellen Forschung vor allem zwei miteinander verbundene Bereiche herauskristallisieren, in denen intensiv mit der Satzlänge gearbeitet wird. Es ist dies der Bereich der quantitativen Klassifikation von Texten (vgl. Mistrík, 1973; Pieper, 1979; Karlgren und Cutting, 1994; Bolton und Roberts, 1995). Gemeinsam ist diesen Untersuchungen, dass in der Regel versucht wird, aufgrund der (durchschnittlichen) Satzlänge und einer ganzen Reihe weiterer quantitativ erfassbarer Textmerkmale bestimmte Textsorten, Genre-Gruppen, Diskurs-Stile u.ä. zu identifizieren. Einen Spezialbereich innerhalb der Textklassifikation nimmt die Frage ein, inwiefern die Satzlänge als eine Teilgröße eines "linguistischen Fingerabdruckes" zu verstehen ist und somit als Merkmal bei der Bestimmung der Autorschaft herangezogen werden kann (vgl. Yule, 1939; Smith, 1983; Kjetsaa, 1984; Holmes, 1994).

[1] Nicht weiter vorgestellt werden soll die Diskussion der Satzlänge im Bereich von psycholinguistischen Untersuchungen. In diesen wird – ausgehend von der als Indikator der grammatikalisch-syntaktischen Komplexität verstandenen Satzlänge – versucht, selbige als Grad der "Lesbarkeit" bzw. "Textverständlichkeit" qualitativ zu interpretieren (vgl. dazu Teigeler, 1968; Flesch, 1948; Tuldava, 1993).

Ad 2. Parallel zu den soeben genannten Fragestellungen rückt die Diskussion in den Vordergrund, inwiefern sich die Verteilung von Satzlängen in adäquater Weise durch theoretische Verteilungsmodelle beschrieben werden kann. Derartige[2] Arbeiten wurden bereits in den 40ern Jahre des 20. Jhd.s durchgeführt (vgl. Yule, 1939; Williams, 1940) und werden bis in die Gegenwart hinein diskutiert (so z.B. von Sichel (1974); Sigurd et al. (2004) u.a.). Einen grundlegenden Neuansatz in der Modellierung der Satzlängenverteilung liefert Altmann (1988b,a), der die Satzlängenverteilung in einen synergetisch-linguistischen Kontext (vgl. Köhler, 1986) stellt und dabei konkret systeminterne und systemexterne Einflussfaktoren in Betracht zieht. Bei einer derartigen Modellierung steht vor allem die Frage im Vordergrund, inwiefern der Autor, die Textsorte, der Stil, der Rezipient u.ä. einen Einfluss auf die Adäquatheit eines theoretischen Verteilungsmodells haben.

Satzlängenforschungen sind aus den genannten Gründen in der Gegenwart nach wie vor ein aktuelles und wichtiges interdisziplinäres Thema. Im Gegensatz zu früheren Untersuchungen werden gegenwärtige Satzlängen-Analysen jedoch in der Regel (a) korpusbasiert und (b) computergestützt durchgeführt. Gerade im Hinblick auf den zuletzt genannten Punkt ist jedoch in erster Linie eine elementare Rahmenbedingung derartiger Untersuchungen zu hinterfragen: nämlich, inwiefern operationale Kriterien einer Satzdefinition postuliert werden können, die überhaupt für eine computerbasierte Analyse zugänglich sind. Aus diesem Grunde wird im Folgenden als erstes (a) näher auf die Frage einer möglichen automatisierten Bestimmung der Satzlänge in Texten einzugehen sein (wobei wir uns hier exemplarisch auf slowenische Prosatexte beschränken), um dann in einem zweiten Schritt (b) zu prüfen, inwiefern sich signifikante Unterschiede bei der quantitativen Auswertung der Satzlänge auf der Ebene der Mittelwerte, der Schiefe und der Kurtosis ergeben, und dann in einem dritten Schritt (c) zu analysieren, inwiefern die Wahl einer bestimmten Satzdefinition als ein Einflussfaktor der theoretischen Modellierung von Satzlängen anzusehen ist.

3 Automatisierte Bestimmung der Einheit ‚Satz'

Die Frage, was unter einem ‚Satz' zu verstehen ist, stellt in der Satzlängenforschung einen durchaus prominenten Aspekt dar. Die für verschiedene theoretische und praktische Fragestellungen notwendige Definition von ‚Satz' ist allerdings keineswegs einheitlich. Dabei sollte es eigentlich als Gemeinplatz anzusehen sein, dass für jegliche formale und quantitative Arbeiten eine stringente und intersubjektiv nachvollziehbare Bestimmung der jeweiligen Untersuchungseinheit(en) – im gegebenen Fall also des ‚Satzes' und der für die Bestimmung der Satzlänge notwendigen Maßeinheiten – notwendig ist; vgl. dazu die theoretischen Implikationen der Quantifizierung und Formalisierung von Altmann und Lehfeldt (1980).

[2] Eine ausführliche Diskussion zu adäquaten Verteilungsmodellen der Häufigkeitsverteilungen der Satzlänge findet sich in Kelih und Grzybek (2004)

Während aus systemlinguistischer Sicht die Messung der Satzlänge in der Anzahl der Teilsätze pro Satz plausibel erscheint, erweist sich aus pragmatischen Gründen die Berechnung der Wortlänge in der Anzahl der Wörter pro Satz als ein mindestens ebenso praktikabler Weg, weil in diesem Fall keine zusätzlichen syntaktischen Analysen notwendig sind. In Anbetracht der Tatsache, dass offenbar beide Vorgangsweisen zum Nachweis sprachlicher Regularitäten führen, wird in der vorliegenden Arbeit davon ausgegangen, dass die Anzahl der Wörter pro Satz eine durchaus sinnvolle Maßeinheit ist. Das ‚Wort' seinerseits wird dabei auf orthographischer Ebene definiert und als eine durch eine Leerstelle abgegrenzte Einheit des Textes verstanden – zur Diskussion alternativer Wortdefinitionen und deren Auswirkung auf quantitative Untersuchungen (vgl. Antić et al., 2006).

Im vorliegenden Text soll es also weniger um eine Untersuchung der Maßeinheit gehen, in welcher die Länge eines Satzes zu berechnen ist, sondern vielmehr um die Definition von ‚Satz'. Im Grunde genommen geht es also um die primäre Frage, ob und wie sich aufgrund von klar definierten Kriterien die (linguistische) Einheit ‚Satz' in Texten und Korpora automatisiert bestimmen lässt. Aus dieser primären Fragestellung leiten sich zwei weitere ab, die mit der Auswirkung der jeweiligen Entscheidung in Zusammenhang stehen, nämlich:

1. Sind in Abhängigkeit von der Wahl einer bestimmten Satzdefinition statistisch signifikante Unterschiede in der durchschnittlichen Satzlänge, der Schiefe und der Kurtosis der Satzlängenverteilungen in den untersuchten Texten zu beobachten?

2. Hat die Wahl der Satzdefinition einen Einfluss auf postulierte theoretische Wahrscheinlichkeitsverteilungen?

Zumindest innerhalb der quantitativen bzw. statistischen Linguistik besteht relativ große Einigkeit darüber, dass Interpunktionszeichen die Funktion haben, einen schriftlichen Text in Einheiten zu gliedern. Daher können Interpunktionszeichen wie der Satzendepunkt herangezogen werden, um Satzgrenzen in Texten zu bestimmen. Eine bekannte Definition, die zahlreichen konkreten analytischen Arbeiten zugrunde liegt, ist die von Bünting und Bergenholtz (1995, p. 27); ihnen zufolge sind Sätze in Texten "solche Einheiten, die durch Satzzeichen eingegrenzt sind". In einer ersten Annäherung handelt es sich hier um eine sinnvolle Operationalisierung. Ungeachtet allfälliger Fragen der Interpunktion in Abhängigkeit von Editionsproblemen[3] ist im hier gegebenen Stelle die Frage nach dem Inventar von Interpunktionszeichen, welche als Markierung eines expliziten Satzendes herangezogen werden, weitaus wichtiger. Dabei liegt es auf der

[3] Vor der Diskussion der relevanten satzabschließenden Zeichen ist kurz auf die Rolle der Interpunktion in schriftlichen Texten und auf allfällige Editionsänderungen einzugehen. Als problematisch zu sehen ist, dass die Interpunktion aufgrund von Editionseingriffen nicht immer der Autorenintention (vgl. Wake, 1957, p. 334) entspricht. So zeigte Janson (1964) in einer akribischen Studie, dass unterschiedliche Editionen in der Setzung der Interpunktionszeichen stark divergieren. Dies heißt aber auch, dass beispielsweise die durchschnittliche Satzlänge aufgrund unterschiedlicher Editionen in einem beträchtlichen Maß divergiert. Im Grunde genommen ist damit jedoch nur ein Teilproblem jeder empirischen Untersuchung angesprochen und impliziert eine eindeutige Darlegung der jeweils verwendeten Textbasis.

Hand, dass eine etwaige Satzdefinition nicht zwangsläufig allgemein sprachübergreifend gültig sein muss.

Es können jedoch auch innerhalb einer Sprache durchaus verschiedene Definitionen des interpunktorischen Satzes verwendet werden. Während in zahlreichen, vor allem auch früheren Arbeiten zum Deutschen wie z.B. derjenigen von Weiß (1968, p. 55), ausschließlich der Punkt, das Frage- und das Ausrufezeichen als satzabschließende Interpunktionszeichen gewertet wurden, finden sich in neueren Arbeiten wie z.B. bei Niehaus (1997, p. 221) differenziertere Kriterien als Grundlage einer Satzdefinition: "Als satzabschließend gelten die folgenden Interpunktionszeichen: der Punkt, das Fragezeichen, das Ausrufezeichen. Der Doppelpunkt nimmt eine Sonderstellung ein, denn er wird nur dann als satzabschließendes Zeichen gewertet, wenn das erste Graphem des folgenden Wortes groß geschrieben wird". Wie zu sehen ist, wird zur Definition des Satzes bzw. des Satzendes jeweils ein anderes Inventar an Interpunktionszeichen herangezogen und zum Teil an bestimmte Kontextbedingungen gekoppelt. Als eine Schlussfolgerung daraus ergibt sich in weiterer Folge natürlich, dass solche Definitionsunterschiede nicht ohne Auswirkung auf quantitative Berechnungen bleiben. Und um diese Frage soll es im vorliegenden Text gehen; zunächst aber gilt es, bevor wir derartige Definitionen automatisch auf die zu analysierenden slowenischen Texte übertragen, die Spezifik der Setzung von Interpunktionszeichen in den Texten der erwähnten Sprache aufzuzeigen.

3.1 Satzdefinition und automatische Bestimmung der Satzlänge im Slowenischen

Nach den Regeln der slowenischen Orthographie kommt dem Punkt die zentrale Funktion zu, das Ende von Sätzen zu markieren. Das Frage- und Ausrufezeichen dient, wie in anderen Sprachen auch, zur Kennzeichnung von Frage- und Ausrufesätzen sowie von Interjektionen (vgl. Pravopis, 1990, p. 38ff.). Unter Berücksichtigung der Wichtigkeit der erwähnten Interpunktionszeichen bei der Textgliederung lautet somit eine erste – in der Praxis durchaus gängige und bei entsprechenden Analysen angewandte Arbeitsdefinition:

Definition 1 *Ein Satz ist eine durch Punkt, Frage- und Ausrufezeichen abgegrenzte Einheit des Textes*

Es ist offensichtlich, dass eine solche Definition nur funktionieren kann und nur dann überhaupt sinnvoll ist, wenn die Datenbasis vor der automatischen Analyse der Texte ‚manipuliert' wird, unter Berücksichtigung der Tatsache, dass der Punkt in der Funktion als Kennzeichnung von Abkürzungen (Beispiele: *c.kr., sv.*) und in der Form von Aufzählungen (Beispiel: *To je Stritarjevo ... tam*) nicht als in satzabschließender Position vorkommend gezählt wird. Während die Problematik der Abkürzungen sich gegebenenfalls durch eine automatisierte Auflösung in Form eines Abkürzungsverzeichnisses überwinden lässt (was aufgrund der starken morphologischen Variationen im Slowenischen bei weitem nicht unproblematisch ist), bleibt die Schwierigkeit des Umgangs mit durch mehrere Punkte gekennzeichneten Aufzählungen bestehen. Noch problematischer an der Definition 1 ist jedoch, dass der Punkt, das Fragezeichen und das Ausrufezeichen

nicht in allen Fällen unbedingt das Ende eines Satzes kennzeichnen müssen. Aus diesem Grund wird im folgenden gezeigt, dass auch eine alternative Zählung möglich und sinnvoll ist.

So bietet es sich in Anlehnung an die grundsätzlichen Überlegungen von Grinbaum (1996) zur Automatisierung von Satzlängenuntersuchungen beispielsweise an, den Großbuchstaben als weiteres satzabgrenzendes Zeichen in eine formal bestimmbare Satzdefinition einzubauen. Wenn auch der Großbuchstabe im Slowenischen primär zur Kennzeichnung von Eigennamen, geographischen Bezeichnungen und ähnlichem dient, liegt eine weitere zentrale Funktion des Großbuchstabens darin, den Anfang von Texten, Absätzen und einzelnen Sätzen zu markieren. In dieser Funktion als Gliederungsmerkmal von Texten können Großbuchstaben bei der Bestimmung von Satzgrenzen herangezogen werden (vgl. Grinbaum, 1996, p. 454). Somit sind die Interpunktionszeichen [.], [. . .], [?] und [!] in Kombination mit einem Großbuchstaben am Anfang des nächstfolgenden Satzes eindeutig als satzabschließend zu identifizieren. Da jedoch den erwähnten Interpunktionszeichen nicht in allen Fällen ein Buchstabe folgen muss (Textende, Absatzende), gelangt man zu der folgenden alternativen operationalen Satzdefinition 2.

Definition 2 *Als Satzendezeichen gelten [.], [. . .], [?] und [!], es sei denn, ein Kleinbuchstabe ist das erste Graphem des nächsten Wortes.*

Ohne Frage ist es möglich, mit beiden aufgezeigten Satzdefinitionen, den Satz und damit auch die Satzlänge automatisiert bestimmen zu können. Die beiden vorgestellten Satzdefinitionen stellen den Ausgangspunkt für unsere empirische Untersuchung der Satzlängenverteilung in slowenischen Texten dar. In weiterer Folge wird dann zu prüfen sein, ob die Anwendung von unterschiedlichen Satzdefinitionen Auswirkung auf statistische Kenngrößen wie Mittelwert, Schiefe und Kurtosis hat. A priori ist bei der Anwendung der Satzdefinition 2 zu erwarten, dass sich die absolute Anzahl der Sätze gegenüber Satzdefinition 1 verringert. Der Grund dafür ist darin zu sehen, dass auch Ausrufe- und Fragesätze innerhalb eines Satzgefüges als vollwertige Sätze betrachtet werden, wie das folgende Beispiel aus Text #1[4] zeigt: *"Kako se vam godi, oče – sedaj, ko ste za starega?" vprašal sem ga s smehom*. Nach Satzdefinition 1 werden hier zwei Sätze mit zehn und fünf Wörtern gezählt, während aufgrund von Satzdefinition 2 ein einziger Satz mit 15 Wörtern ausgewertet wird. Ähnlich wird beispielsweise die folgende Sequenz aus Text #3.2 je nach Satzdefinition entweder als ein Satz mit neun Wörtern oder als zwei Sätze mit fünf und vier Wörtern ausgezählt.: *"Kaj jo je zbodlo, ženščuro?" se je začudil Jernej [. . .]"*

3.2 Textkorpus der slowenischen literarischen Texte

Als Basis für die empirische Untersuchung dient ein Korpus slowenischer Prosatexte. Dieses setzt sich aus sechs Kurzromanen bzw. Kurzgeschichten (slowenisch: povest),

[4] Die hier genannten arabischen Zahlen bezeichnen die im Kapitel 3.2 eingeführte Nummerierung der analysierten Texte.

im Folgenden als ‚Kurzerzählungen' bezeichnet, von vier verschiedenen slowenischen Autoren aus dem 19. Jhd. zusammen (vgl. Tabelle 1).

Text	Autor	Titel
# 1	J. Kersnik	*Mačkova očeta*
# 2	J. Kersnik	*Ponkrčev oča*
# 3	I. Cankar	*Hlapec Jernej in njegova pravica*
# 4	J. Jurčič	*Nemški Valpet*
# 5	F. Levstik	*Pokljuk*
# 6	F. Levstik	*Martin Krpan*

Tabelle 1: Das Korpus slowenischer Texte.

Diese Texte werden im Folgenden mit den entsprechenden Nummern (Text #1 bis Text #6) bezeichnet und als solche analysiert. Zum Zwecke der Kontrolle der Datenbasis – zur Frage der Homogenität in quantitativen Untersuchungen vgl. Altmann (1992) bzw. Orlov (1982) – werden diese Texte jedoch nicht nur jeweils einzeln als komplexe Gesamttexte analysiert, sondern auf zwei weitere Arten und Weisen: Zum einen werden die genannten sechs Texte zu einem Gesamtkorpus zusammengefügt, so dass sich eine umfangreichere Textmischung ergibt; dieses Gesamtkorpus soll im folgenden bedingt als "Text #7" bezeichnet werden (vgl. Tabelle 2). Zum anderen ergibt sich aufgrund der Tatsache, dass die Texte #2 und #3 jeweils aus mehreren Kapiteln bestehen, die Option, diese einzelnen Kapitel jeweils als homogene Texte zu verstehen und getrennt zu analysieren; in diesem Fall haben wir es mit den Texten #8 bis #28 zu tun (vgl. Tabelle 3). Auf diese Art und Weise lässt sich die Qualität des den Analysen zugrunde gelegten Datenmaterials zuverlässig kontrollieren. In Übersicht lässt sich nunmehr die Satzlänge auf den folgenden drei Ebenen bestimmen:

1. Auf der ersten Ebene werden die genannten Kurzerzählungen (vgl. Tabelle 1) zu einem vollständigen Korpus zusammengefasst. Das Korpus, welches in Hinsicht auf die involvierten Textsorten als homogen zu bezeichnen ist, gibt die Möglichkeit zu prüfen, inwiefern eine Korpusanalyse gegebenenfalls eine andere Satzlängenverteilung aufweist als die Analyse der einzelnen Texte. Insgesamt besteht das Korpus aus 39016 Wörtern; gemäß Satzdefinition 1 werden insgesamt 2938 Sätze ausgezählt, während bei Anwendung von Satzdefinition 2 insgesamt 2758 Sätze zu verbuchen sind (also ein Unterschied von immerhin ca. 6.5%). Die weiteren Werte, vor allem auch die mittlere Satzlänge, sind im Einzelnen der Tabelle 2 zu entnehmen.

2. Auf der zweiten Ebene werden die sechs Kurzerzählungen als komplexe Texte aufgefasst; anzumerken ist, dass dabei eine textinterne, von den Autoren selbst vorgenommene Kapitelgliederung nicht beachtet wird. Insgesamt handelt es sich

Satzdefinition	Text	Sätze	Wörter	$\bar{x}$
1	#7	2938	39016	13,28
2		2758	39016	14,15

Tabelle 2: Quantitative Angaben zum Textkorpus.

also um sechs unterschiedliche Kurzerzählungen von vier verschiedenen Autoren, wobei Text #3 mit insgesamt 1383 Sätzen den größten Umfang aufweist. Unter dieser Voraussetzung zeichnen sich die Texte durch die in Tabelle 3 zusammengefassten Charakteristika aus.

		Satzdefinition 1		Satzdefinition 2	
Text	Wörter	Sätze	$\bar{x}$	Sätze	$\bar{x}$
# 1	1597	118	13,53	109	14,65
# 2	2178	191	11,40	165	13,20
# 3	18407	1493	12,33	1383	13,31
# 4	7971	585	13,63	561	14,21
# 5	3181	170	18,71	169	18,82
# 6	5682	381	14,91	371	15,32

Tabelle 3: Quantitative Angaben zu den komplexen Texten.

3. Auf der dritten Ebene werden die Texte unter Berücksichtigung der von den Autoren selbst vorgenommen Kapitelgliederung jeweils individuell untersucht. Im Detail weist der Text von Janko Kersnik *Ponkrčev oča* drei Kapitel auf; Ivan Cankars *Hlapec Jernej in njegova pravica* besteht aus 18 Einzelkapiteln; insgesamt stehen also 21 Einzelkapitel für die Analysen zur Verfügung. Tabelle 4 resümiert die wesentlichen Charakteristika der Einzeltexte.

Durch die differenzierte Analyse auf drei unterschiedlichen Ebenen ergeben sich insgesamt 28 Datensätze, für welche die Satzlänge unter Anwendung beider Satzdefinitionen bestimmt werden kann. Auf der Basis dieser Texte lässt sich nunmehr – neben der Frage des Einflussfaktors der Satzdefinition – auch die Frage der Datenhomogenität kontrollieren.

3.3 Statistische Vergleiche der durchschnittlichen Satzlänge

Als erstes wird in den sechs komplexen Texten (vgl. Tabelle 1) nach Satzdefinitionen 1 und 2 die in der Anzahl der Worte gemessene Satzlänge automatisiert bestimmt. Es zeigt sich in diesem ersten Schritt, dass aufgrund der Satzdefinition 2 in allen Texten

		Satzdefinition 1		Satzdefinition 2	
Text	**Wörter**	**Sätze**	$\bar{x}$	**Sätze**	$\bar{x}$
# 8	895	76	11,78	68	13,16
# 9	523	44	11,89	38	13,76
# 10	760	71	10,70	59	12,88
# 11	602	47	12,81	43	14,00
# 12	977	92	10,62	82	11,91
# 13	1038	90	11,53	85	12,21
# 14	796	61	13,05	57	13,96
# 15	809	81	9,99	80	10,11
# 16	890	81	10,99	75	11,87
# 17	973	79	12,32	71	13,70
# 18	1473	120	12,28	107	13,77
# 19	939	65	14,45	60	15,65
# 20	1134	120	9,45	113	10,04
# 21	937	80	11,71	75	12,49
# 22	1203	80	15,04	80	15,04
# 23	1583	126	12,56	119	13,30
# 24	956	61	15,67	53	18,04
# 25	1388	107	12,97	98	14,16
# 26	1203	84	14,32	77	15,62
# 27	1203	98	12,28	87	13,83
# 28	303	21	14,43	21	14,43

Tabelle 4: Quantitative Angaben zu den Einzeltexten.

eine geringere absolute Anzahl an Sätzen ($N_1 < N_2$) ausgezählt wird. Dieser Befund deutet darauf hin, dass aufgrund der Satzdefinition 2 der Text in größere (d.h. längere) Einheiten eingeteilt wird; dementsprechend ändert sich auch die in den Texten berechnete durchschnittliche Satzlänge. Beispielsweise ergibt sich für den komplexen Text #2 aufgrund der Satzdefinition 1 eine mittlere Satzlänge von $\bar{x} = 11{,}40$ Wörtern pro Satz, während auf der Grundlage von Satzdefinition 2 die mittlere Satzlänge $\bar{x} = 13{,}20$ beträgt. Man sieht somit, dass die beiden Satzdefinitionen sich offensichtlich unmittelbar und massiv auf die durchschnittliche Satzlänge auswirken.

Es stellt sich nun die Frage, inwiefern diese Beobachtung an allen Texten nachzuweisen ist, oder ob die soeben angesprochenen Unterschiede als ein Einzelfall zu betrachten sind. Entsprechend werden für alle slowenischen Datensätze die durchschnittlichen Satzlängen aufgrund der beiden angeführten Satzdefinitionen berechnet, und die sich ergebenden Satzlängenunterschiede auf statistische Signifikanz geprüft; die einzelnen Werte finden sich in der Tabelle 5.

Wie zu sehen ist, unterscheiden sich lediglich zwei der 28 Stichproben nicht im Hinblick auf die durchschnittliche Satzlänge (Text #22 und Text #28). Bei allen anderen Datensätzen wirkt sich die Satzdefinition auf den Mittelwert aus, wobei sich natürlich die Frage nach der Signifikanz des Unterschiedes stellt. Für einen statistischen Vergleich der Mittelwerte unter beiden Bedingungen – die wir als unabhängige Stichproben ansehen wollen – ist es üblich, den sog. Zweistichproben *t*-Test durchzuführen, der auch in der Linguistik breite Verwendung gefunden hat. Dieser *t*-Test ist u.a. in der Satzlängenforschung angewendet worden, um zu testen, ob sich zwei Mittelwerte aus zwei unabhängigen Stichproben auf einem festgelegten Niveau unterscheiden (vgl. Grzybek, 2000, p. 446). Geprüft wird also die Nullhypothese, dass sich zwei Mittelwerte auf einem festgelegten Signifikanzniveau ($\alpha = 0{,}05$ und fakultativ $\alpha = 0{,}01$) nicht unterscheiden. Die Berechnungsverfahren des *t*-Werts unterscheiden sich geringfügig in Abhängigkeit davon, ob die Varianzen beider Stichproben homogen sind oder nicht, was mit Hilfe der sog. Levene-Statistik berechnet werden kann; in unserem Fall zeigt die Levene-Statistik, dass alle Stichproben Homogenität der Varianzen aufweisen. Als Ergebnis dieses Mittelwertvergleichs stellt sich heraus, dass sich lediglich zwei der 28 Stichproben – nämlich Text #3, der mit Abstand der längste der Texte ist, sowie Text #7, das gesamte Korpus) – auf dem 1%-Niveau signifikant unterscheiden. Ein weiterer Text (Text #2) kommt hinzu, wenn man das Signifikanzniveau bei 5% ansetzt; hierbei handelt es sich interessanterweise um einen solchen Text, der einen hohen Anteil an Frage- und Ausrufesätzen aufweist.

Bevor man jedoch zu vorschnellen Interpretationen gelangt, gilt es folgendes zu berücksichtigen: Voraussetzung für die Durchführung des *t*-Tests ist jedoch, dass die Werte beider Stichproben normalverteilt sind; dies ist in unseren beiden Stichproben – wie entsprechende Tests zeigen – allerdings nicht der Fall. Da somit die Anwendung des *t*-Tests aufgrund der Verletzung der vorausgesetzten Normalverteilung nicht zulässig ist, muss der sog. *U*-Test nach Mann/Whitney zum Einsatz kommen. Ähnlich wie der *t*-Test dient auch er dem Vergleich von zwei Stichproben hinsichtlich ihrer zentralen Tendenz, allerdings können hier die Werte beliebig verteilt sein oder Ordinalniveau aufweisen. Tabelle 5 enthält in den beiden letzten Spalten die entsprechenden *z*-Werte sowie die ihnen entsprechenden Wahrscheinlichkeiten.

Wie der Tabelle 5 zu entnehmen ist, führt die zulässige Anwendung des *U*-Tests in unserem Fall im Wesentlichen zu ein und denselben Ergebnissen wie der *t*-Test: Auf dem 1%-Niveau gibt es signifikante Abweichungen nur beim gesamten Korpus (Text #7), sowie bei den Texten #2 und #3; hinzu kommt lediglich Text #10, wenn man die Signifikanzschwelle auf 5% senkt.

Damit lässt sich, dieses erste Ergebnis zusammenfassend, sagen, dass die beiden vorgestellten Satzdefinitionen in der Tat zu signifikanten Unterschieden führen können. Bemerkenswert ist dabei, dass dies in erster Linie beim Gesamtkorpus (Text #7) und bei dem längsten Text #3 der Fall ist; insofern scheint es plausibel anzunehmen, dass eine bestimmte Anzahl von Beobachtungen (d.h. eine gewisse Textlänge) notwendig ist, damit überhaupt signifikante Unterschiede zum Tragen kommen können. Andererseits scheint es aber durchaus auch textspezifische bzw. textsortenspezifische Einflüsse zu geben, die in einem signifikanten Mittelwertunterschied resultieren können.

	n		$\bar{x}$		s		t-Test			Levene-Test		U-Test	
Text	n_1	n_2	$\bar{x}_1$	$\bar{x}_2$	s_1	s_2	t	FG	p		p	z	p
#7	2938	2758	13,28	14,15	9,82	9,79	-3,33	5694	**0,001**	0,168	0,682	-4,48	< **0,001**
#1	118	109	13,53	14,65	9,95	10,48	-0,82	225	0,411	0,010	0,921	-0,90	0,3660
#2	191	165	11,40	13,20	8,44	8,30	-2,02	354	**0,044**	0,137	0,712	-2,65	**0,0080**
#3	1493	1383	12,33	13,31	9,00	8,94	-2,93	2874	**0,003**	0,288	0,592	-3,81	< **0,001**
#4	585	561	13,63	14,21	9,58	9,65	-1,03	1144	0,310	0,042	0,837	-1,24	0,2160
#5	170	169	18,71	18,82	11,89	11,87	-0,09	337	0,932	0,000	0,982	-0,10	0,9200
#6	381	371	14,91	15,32	11,62	11,56	-0,48	750	0,635	0,002	0,960	-0,75	0,4530
#8	76	68	11,78	13,16	8,27	7,98	-1,02	142	0,309	0,143	0,706	-1,28	0,2000
#9	44	38	11,89	13,76	7,76	8,24	-1,06	80	0,291	0,077	0,782	-1,31	0,1920
#10	71	59	10,70	12,88	9,06	8,81	-1,38	128	0,170	0,147	0,702	-2,06	**0,0390**
#11	47	43	12,81	14,00	8,66	8,41	-0,66	88	0,510	0,045	0,832	-0,92	0,3560
#12	92	82	10,62	11,91	8,00	8,05	-1,06	172	0,289	0,030	0,862	-1,34	0,1800
#13	90	85	11,53	12,21	8,34	8,35	-0,54	173	0,592	0,008	0,927	-0,73	0,4660
#14	61	57	13,05	13,96	15,31	15,49	-0,32	116	0,747	0,000	0,977	-0,73	0,4680
#15	81	80	9,99	10,11	7,46	7,47	-0,11	159	0,916	0,001	0,979	-0,13	0,8950
#16	81	75	10,99	11,87	7,83	7,75	-0,70	154	0,482	0,046	0,831	-1,01	0,3140
#17	79	71	12,32	13,70	8,27	7,98	-1,04	148	0,289	0,095	0,758	-1,37	0,1710
#18	120	107	12,28	13,77	8,01	8,11	-1,39	225	0,165	0,010	0,919	-1,54	0,1240
#19	65	60	14,45	15,65	7,66	6,92	-0,92	123	0,360	0,741	0,391	-0,90	0,3670
#20	120	113	9,45	10,04	7,17	7,24	-0,62	231	0,536	0,002	0,966	-0,71	0,4790
#21	80	75	11,71	12,49	7,20	6,94	-0,69	153	0,493	0,213	0,645	-0,75	0,4550
#22	80	80	15,04	15,04	11,28	11,28	0,00	158	1,000	0,000	1,000	0,00	1,0000
#23	126	119	12,56	13,30	8,78	8,59	-0,67	243	0,506	0,062	0,804	-0,84	0,3990
#24	61	53	15,67	18,04	11,60	11,46	-1,09	112	0,277	0,015	0,904	-1,55	0,1200
#25	107	98	12,97	14,16	8,33	8,07	-1,04	203	0,301	0,085	0,771	-1,20	0,2290
#26	84	77	14,32	15,62	10,40	10,12	-0,80	159	0,423	0,083	0,773	-1,16	0,2440
#27	98	87	12,28	13,83	7,37	6,98	-1,47	183	0,145	0,223	0,637	-1,53	0,1260
#28	21	21	14,43	14,43	9,35	9,35	0,00	40	1,000	0,000	1,000	0,00	1,0000

Tabelle 5: Vergleich der durchschnittlichen Satzlängen.

Ohne Zweifel ist die Frage nach der zentralen Tendenz einer Stichprobe bzw. nach Unterschieden in der zentralen Tendenz eine der meist gestellten Fragen im Rahmen von Satzlängenforschungen. Der Grund für die Beliebtheit dieser Fragestellung dürfte darin zu sehen sein, dass es hier um die Spezifik individueller Texte geht. In einem breiteren Kontext jedoch scheint eine in eine andere Richtung zielende Frage von mindestens ebenso großer Bedeutung; diese geht ebenfalls von bestimmten Charakteristika der Häufigkeitsverteilung aus, fragt jedoch in erster Linie nach einem allfällig gemeinsamen Profil der Verteilungen.

3.4 Statistischer Vergleich von Schiefe und Kurtosis

Konkret bietet sich zur Untersuchung der zuletzt genannten Fragestellung die Analyse von Schiefe (γ_1) und Kurtosis (γ_2) der Häufigkeitsverteilung an. Mit diesen beiden Maßen wird angegeben, in welchem Maße eine Verteilung im Vergleich zur Normalverteilung links- oder rechtsverschoben bzw. höher oder niedriger als diese liegt: Im Falle einer linkssteilen Verteilung spricht man von einer positiven Schiefe (d.h. $\gamma_1 > 0$), im Fall einer steilgipfligen Verteilung von einem positiven Exzeß ($\gamma_2 > 0$). Im gegebenen Fall interessiert zwar nicht in erster Linie der Vergleich zur Normalverteilung, wohl aber

die Frage, ob es signifikante Unterschiede in Schiefe und/oder Kurtosis in Abhängigkeit von der Satzdefinition gibt.

Tabelle 6 repräsentiert die Werte für Schiefe und Kurtosis für beide Satzdefinitionen.

	Schiefe		Vergleich Schiefe		Kurtosis		Vergleich Kurtosis	
Text	γ_1 (1)	γ_1 (2)	z	p	γ_2 (1)	γ_2 (2)	z	p
# 7	1,9490	1,9760	-0,2811	0,7787	8,0370	8,2700	-0,2465	0,8053
# 1	1,4082	1,6671	-1,6975	0,0896	3,3184	4,3692	-1,5431	0,1228
# 2	1,2154	1,1316	1,1147	0,2650	1,2154	1,0473	0,7077	0,4791
# 3	2,0300	2,0545	-0,1486	0,8818	9,8870	10,4402	-0,3480	0,7278
# 4	1,6374	1,6401	-0,0419	0,9666	3,6903	3,5506	0,3954	0,6925
# 5	1,4115	1,4196	-0,0897	0,9286	3,0785	3,0960	-0,0469	0,9626
# 6	2,2759	2,3054	-0,1176	0,9064	10,6192	10,8907	-0,1305	0,8962
# 8	0,9217	0,8320	0,7574	0,4488	0,5303	0,5648	-0,0993	0,9207
# 9	1,1929	1,1195	0,6075	0,5435	0,8089	0,3235	1,2163	0,2239
# 10	1,4996	1,4117	0,8015	0,4228	1,9920	1,7811	0,4494	0,6531
# 11	1,5894	1,6429	-0,4517	0,6515	2,7709	2,8772	-0,1649	0,8690
# 12	1,1366	1,1049	0,3240	0,6400	0,8356	1,2083	-1,1318	0,2577
# 13	1,6953	1,6086	0,4955	0,6203	4,5089	4,3323	0,2159	0,8291
# 14	3,4854	3,4589	0,1139	0,9093	14,8248	14,3772	0,1739	0,8619
# 15	1,2904	1,2632	0,2246	0,8223	1,8174	1,7635	0,1255	0,9001
# 16	1,3642	1,3839	-0,2066	0,8363	1,4508	1,4708	-0,0489	0,9610
# 17	1,1457	1,1000	0,4475	0,6545	1,1984	1,2190	-0,0595	0,9525
# 18	0,8086	0,7935	0,1810	0,8564	0,3222	0,8042	-1,9601	0,0500
# 19	0,3429	0,5021	-1,8844	0,0595	-0,5813	-0,4817	-0,5844	0,5590
# 20	1,3580	1,2329	0,7955	0,4263	2,9518	2,6195	0,6302	0,5285
# 21	0,5450	0,5093	0,4039	0,6863	0,0256	0,1825	-0,8115	0,4171
# 22	2,4774	2,4774	0,0000	1,0000	10,5769	10,5769	0,0000	1,0000
# 23	1,2186	1,2452	-0,1933	0,8468	2,2737	2,4774	-0,3469	0,7287
# 24	2,2453	2,2475	-0,0115	0,9908	7,7227	7,7275	-0,0052	0,9958
# 25	0,6540	0,5516	1,5894	0,1120	-0,3293	-0,3797	0,3800	0,7039
# 26	2,0087	2,1433	-0,6821	0,4952	6,2466	6,9015	-0,6421	0,5208
# 27	0,5747	0,4985	1,1295	0,2587	-0,1940	-0,1980	0,2662	0,9788
# 28	0,8701	0,8701	0,0000	1,0000	0,5659	0,5659	0,0000	1,0000

Tabelle 6: Kennwerte zur Schiefe und Kurtosis.

Das Maß der Schiefe (γ_1) ist hier nach der üblichen Formel 1 berechnet:

$$\hat{\gamma}_1 = \frac{m_3}{s^3}, \tag{1}$$

wobei $s^2 = \frac{N}{N-1} \sum_{i=1}^{k} (i - \bar{x})^2 \, p_i$.

Wie der Tabelle 6 zu entnehmen – und wie nicht anders zu erwarten – ist das Maß der Schiefe in allen Texten unter beiden Bedingungen jeweils positiv. Um nun das Ausmaß der Schiefe für beide Satzdefinitionen miteinander zu vergleichen, ist es notwendig, gemäß der Formel 2 die Maße für die Schiefe und die Varianz der Schiefe unter beiden

Bedingungen – in der Formel als (a) und (b) gekennzeichnet – zueinander in Beziehung zu setzen.

$$z_1 = \frac{\hat{\gamma}_{1(a)} - \hat{\gamma}_{1(b)}}{\sqrt{Var\left(\hat{\gamma}_{1(a)}\right) + Var\left(\hat{\gamma}_{1(b)}\right)}} \tag{2}$$

Während die Varianz der Schiefe üblicherweise unter Annahme der Normalverteilung der Werte nach der Formel 3 berechnet wird,

$$Var\left(\hat{\gamma}_1\right) = \frac{6N \cdot (N-1)}{(N-2) \cdot (N+1) \cdot (N+3)}, \tag{3}$$

ist bei fehlender (bzw. nicht anzunehmender) Normalverteilung die Berechnung nach Lewis und Orav (1989) etwas komplizierter gemäß Formel 4 vorzunehmen:

$$Var\left(\hat{\gamma}\right) = \frac{(N-1)\, L\left(\hat{\gamma}\right)}{4\,(N-2)^2\,(s)^{10}} \tag{4}$$

wobei $L\left(\hat{\gamma}\right) = 4s^4m_6 - 12s^2m_3m_5 - 24s^6m_4 + 9\left(m_3\right)^2 m_4 + 35s^4\left(m_3\right)^2 + 36s^{10}$.

Als Ergebnis der entsprechenden Vergleiche stellt sich heraus, dass sich in allen Fällen die Schiefe nicht signifikant in Abhängigkeit von der jeweiligen Satzdefinition unterscheidet. Dasselbe gilt für die Kurtosis (γ_2), deren Werte ebenfalls der Tabelle 6 zu entnehmen sind. Die Kurtosis wird nach der üblichen Formel 5 berechnet:

$$\hat{\gamma}_2 = \frac{m_4}{s^4} - 3 \tag{5}$$

Zur Prüfung, ob sich das Maß der Kurtosis für beide Satzdefinitionen unterscheidet, ist es abermals notwendig, die Maße für die Kurtosis und die Varianz der Kurtosis unter beiden Bedingungen – in der Formel wiederum als (a) und (b) gekennzeichnet – zueinander in Beziehung zu setzen, und zwar gemäß der Formel 6.

$$z_2 = \frac{\hat{\gamma}_{2(a)} - \hat{\gamma}_{2(b)}}{\sqrt{Var\left(\hat{\gamma}_{2(a)}\right) + Var\left(\hat{\gamma}_{2(b)}\right)}} \tag{6}$$

Die Varianz der Kurtosis berechnet sich hierbei nach der Formel 7:

$$Var\left(\hat{\gamma}_2\right) = \frac{(N-1)^2\left(N^2 - 2N + 3\right)^2 L\left(\hat{\gamma}_2\right)}{(N-2)^2\,(N-3)^2\,(N)^3\,(s)^{12}} \tag{7}$$

wobei $L\left(\hat{\gamma}_2\right) = s^4m_8 - 4s^2m_4m_6 - 8s^4m_3m_5 + 4{m_4}^3 - s^4{m_4}^2 + 16s^2{m_3}^2m_4 + 16s^6{m_3}^2$.

Damit stellt sich insgesamt heraus, dass im Gegensatz zum Befund signifikanter Mittelwertunterschiede (s.o.), in allen Texten die Satzdefinition weder bei der Schiefe noch bei der Kurtosis einen signifikanten Unterschied bewirkt. Dabei gibt es unter beiden

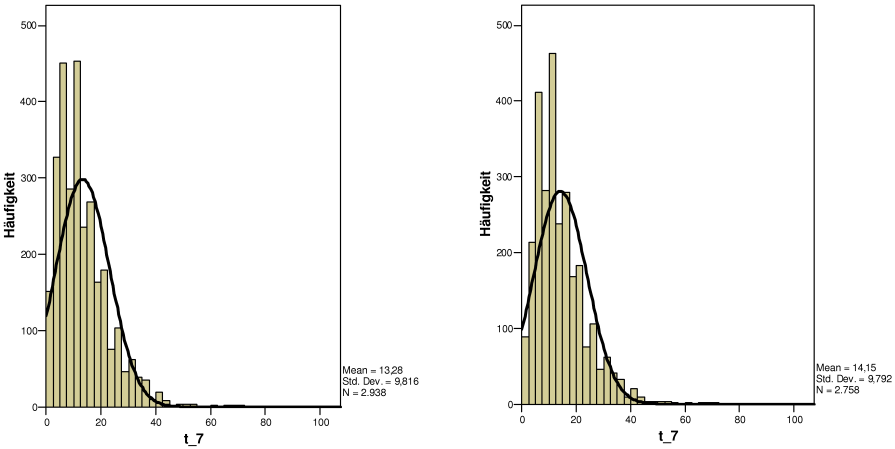


Abbildung 1: Satzlängenhäufigkeitsverteilung für das gesamte Korpus (Text #7) mit eingezeichneter Normalverteilungskurve.

Bedingungen einen positiven linearen Zusammenhang zwischen Schiefe und Kurtosis; dies zeigt der aufgrund der teilweise fehlenden Normalverteilung der Variablen zu berechnende Spearmansche Rangkorrelationskoeffizient ($\rho = 0.98$, für Satzdefinition 1 bzw. $\rho = 0.97$ für Satzdefinition 2, jeweils $p < 0.001$).

Abbildung 1 veranschaulicht das Ergebnis für das Gesamtkorpus (#Text 7), welches auf graphischer Ebene deutlich macht, dass sich das Profil der Häufigkeitsverteilungen in der Tat nicht wesentlich in Abhängigkeit von der Satzdefinition ändert.

Diese Beobachtung leitet allerdings zu der weiterführenden Frage über, inwiefern sich an die Texte ein einheitliches Modell einer diskreten Wahrscheinlichkeitsverteilung anpassen lässt, und inwiefern sich hier entweder im Hinblick auf das Modell insgesamt oder aber in Hinsicht auf die Parameterwerte des entsprechenden Modells Unterschiede in Abhängigkeit von der Satzdefinition ergeben.

4 Theoretische Modellierungen von Satzlängen

Bei der Verfolgung dieser Fragestellung können wir uns auf Ergebnisse einer anderen Studie beziehen, deren Design hier nicht im Detail dargestellt werden muss (vgl. Kelih und Grzybek, 2004). Es ging in dieser Studie um einen anderen möglichen Faktor aus dem Umfeld der Rahmenbedingungen, der möglicherweise die theoretische Modellierung der Satzlängenhäufigkeit beeinflusst: nämlich das zum Zwecke der Datenglättung üblicherweise angewendete Verfahren der Intervallbildungen, das aufgrund der mit Satzlängenhäufigkeiten in der Regel verbundenen hohen Streuung notwendig ist. Ohne

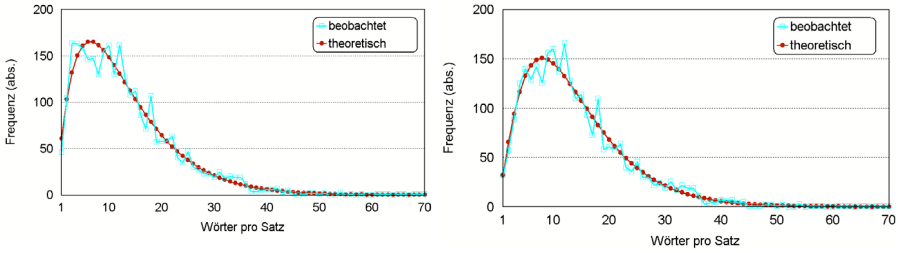


Abbildung 2: Anpassung der negativen Binomialverteilung an die Satzlängenhäufigkeitsverteilung für das gesamte Korpus (Text #7) gemäß Satzdefinition 1 und 2.

auf die Ergebnisse dieser Untersuchung hier im Detail einzugehen, können wir festhalten, dass sich bei den oben genannten Texten (vgl. Tabellen 2-4) die negative Binomialverteilung als durchgehend robustes Modell am besten zur theoretischen Modellierung der Satzlängenhäufigkeiten erweisen hat. Vor diesem Hintergrund erweist es sich nunmehr im Hinblick auf die beiden oben diskutierten Satzdefinitionen als sinnvoll, die allfällige Auswirkung der Satzdefinition auf dieses Modell bzw. dessen konkrete Parameter zu überprüfen.

4.1 Empirische Überprüfung der Verteilung von Satzlängen in slowenischen Texten

Die negative Binomialverteilung ist gegeben als

$$P_x = \left(\begin{array}{c} k+x-1 \\ x \end{array} \right) p^k q^x \qquad x = 0, 1, 2, ... \tag{8}$$

Da Sätze mit 0 Wörtern jedoch prinzipiell ausgeschlossen sind, wird sie korrekterweise in ihrer 0-gestutzten Form angewendet, wie sie in (9) gegeben ist:

$$P_x = \left(\begin{array}{c} k+x-1 \\ x \end{array} \right) \frac{p^k q^x}{1-p^k} \qquad x = 1, 2, ... \tag{9}$$

Die Anpassung dieser Verteilung an die Daten lässt sich mit Spezialsoftware wie dem Altmann-Fitter (2000) computergestützt erreichen (der auch die 0-Stutzung automatisch vollzieht). Dabei werden die Werte für die Parameter nach bestimmten Anfangsschätzungen in iterativen Prozeduren so optimiert, dass der χ^2-Wert als Test für die Güte der Anpassung minimiert wird. Veranschaulichen wir das Vorgehen an Text #7, dem Gesamtkorpus.

Abbildung 2 stellt die beobachteten Häufigkeiten für beide Satzdefinitionen dar, die in Form eines zusammenfassenden Histogramms schon in Abbildung 1 dargestellt wurden (s.o.). Hinzu kommen nun die theoretischen Häufigkeiten, wie sie sich nach Einsetzen der Werte für die Parameter k und p ergeben: Für Satzdefinition 1 betragen die Werte k_1

= 1.97 und p_1 = 0.14, für Satzdefinition 2 betragen k_2 = 2.41 und p_2= 0.16. Das Einsetzen dieser Werte in die obige Formel resultiert in einem Wert von χ^2 = 53.78 bzw. χ^2 = 44.47 für die beiden Satzdefinitionen. Dies entspricht für Satzdefinition 1 einer Wahrscheinlichkeit von P = 0.007 bei 25 Freiheitsgraden, für Satzdefinition 2 von P = 0.0067 (bei 24 Freiheitsgraden). Da üblicherweise Wahrscheinlichkeiten im Falle von $P > 0.05$ als sehr gutes bzw. $P > 0.01$ als gutes Anpassungsergebnis zu werten sind, wäre hier von einer schlechten Anpassung zu sprechen. Allerdings steigt der χ^2-Wert linear mit der Stichprobengröße, weshalb er für große Stichproben Abweichungen schneller als signifikant erscheinen lässt; aus diesem Grund wird in der quantitativen Linguistik im Falle von großen Stichproben statt der Wahrscheinlichkeit P der Wert des Diskrepanzkoeffizienten C herangezogen, der sich als $C = \chi^2 / N$ berechnet (vgl. Grotjahn und Altmann, 1993). Dies bietet sich im Falle von Text #7 bei Stichprobenumfängen von N_1 = 2938 bzw. N_2 = 2758 an und resultiert in als gut zu bezeichnenden Anpassungswerten von C_1 = 0.0183 bzw. C_2 = 0.0161. Fasst man die Satzlängenhäufigkeit in 5er-Intervallen zusammen – wie dies aufgrund der hohen Streuung in der Satzlängenforschung absolut üblich ist – resultiert dies in einer Verringerung der Werte im Falle von Satzdefinition 1 auf C = 0.0121 und bei Satzdefinition 2 auf C = 0.0048, was sich als Indiz für die ausgezeichnete Eignung der negativen Binomialverteilung interpretieren lässt.

Tabelle 7 stellt für alle Texte die Ergebnisse der Anpassung der negativen Binomialverteilung für beide Satzdefinitionen dar; enthalten ist neben dem jeweiligen χ^2-Wert mit den dazugehörigen Freiheitsgraden (*FG*) die diesem Wert entsprechende Wahrscheinlichkeit P, den sich aus der Division von χ^2 und N ergebenden C-Wert sowie die jeweiligen Parameterwerte.

Wie zu sehen ist, erweist sich die negative Binomialverteilung unter der Bedingung beider Satzdefinitionen als geeignetes Modell: Sowohl für Satzdefinition 1 als auch für Satzdefinition 2 lassen sich die Texte durch ein und dasselbe theoretische Verteilungsmodell, nämlich die negative Binomialverteilung, beschreiben. Dabei sind keinerlei Unterschiede in Bezug auf die analysierte Textebene erkennbar, das Modell erweist sich sowohl auf der Ebene der Einzeltexte, als auch der komplexen Texte, als auch des Korpus unter beiden Bedingungen als hervorragend geeignet.

Ungeachtet dessen stellt sich die einen Schritt weiter gehende Frage, ob sich die Wahl der Satzdefinition auf die theoretischen Parameterwerte der Verteilung auswirkt.

5 Parameter der negativen Binomialverteilung

Im Hinblick auf die beiden Parameter der negativen Binomialverteilung (k, p) gibt es eine Reihe möglicher Fragen, was einen eventuellen Einfluss der Satzdefinitionen betrifft. Eine erste Frage zielt darauf, ob die beiden Parameter jeweils eine systematische Verschiebung in Abhängigkeit von der Satzdefinition erfahren.

Wie Abbildung 3 zeigt, gibt es einen positiven linearen Zusammenhang sowohl für k als auch für p in Abhängigkeit von der Satzdefinition: k und p tendieren dazu, unter der Bedingung von Satzdefinition 2 größer zu sein als bei der Anwendung von Satzdefinition 1; die Tendenz ist in beiden Fällen gleichermaßen hoch signifikant, wie der aufgrund

	Satzdefinition 1					*Satzdefinition 2*				
Text	χ^2	*FG*	*P*	*C*	*N*	χ^2	*FG*	*P*	*C*	*N*
# 7	53,78	25	0,0007	**0,0183**	2938	44,47	24	0,0067	**0,0161**	2758
# 1	30,55	35	**0,6829**	0,2589	118	38,04	36	**0,3767**	0,349	109
# 2	36,16	32	**0,2803**	0,1893	191	32,98	33	**0,4680**	0,1999	165
# 3	24,23	5	0,0002	**0,0162**	1493	11,85	4	0,0185	**0,0086**	1383
# 4	58,89	44	**0,0660**	0,1007	585	7,52	5	**0,1845**	0,0134	561
# 5	39,03	41	**0,5583**	0,2296	170	42,15	42	**0,4646**	0,2494	169
# 6	59,87	46	**0,0823**	0,1571	381	15,20	8	**0,0555**	0,041	371
# 8	28,11	26	**0,3529**	0,3699	76	21,74	27	**0,7505**	0,3197	68
# 9	22,06	18	**0,2294**	0,5013	44	21,94	18	**0,2347**	0,5774	38
# 10	18,72	22	**0,6623**	0,2637	71	0,58	3	**0,9002**	0,0099	59
# 11	24,84	21	**0,2541**	0,5286	47	23,19	18	**0,1834**	0,5393	43
# 12	23,39	25	**0,5550**	0,2542	92	13,84	25	**0,9644**	0,1688	82
# 13	23,03	25	**0,5757**	0,2559	90	25,14	26	**0,5113**	0,2957	85
# 14	33,75	24	**0,0892**	0,5533	61	10,41	6	**0,1085**	0,1826	57
# 15	30,30	21	**0,0861**	0,3741	81	27,74	22	**0,1846**	0,3467	80
# 16	24,35	23	**0,3846**	0,3006	81	19,41	23	**0,6772**	0,2588	75
# 17	17,98	25	**0,8434**	0,2275	79	20,78	25	**0,7047**	0,2927	71
# 18	31,22	29	**0,3552**	0,2602	120	38,10	29	**0,1201**	0,3561	107
# 19	3,58	7	**0,8265**	0,0551	65	4,08	6	**0,6664**	0,0679	60
# 20	18,97	13	**0,1241**	0,1581	120	41,17	28	**0,0518**	0,3643	113
# 21	35,37	24	**0,0631**	0,4421	80	23,55	17	**0,1321**	0,3141	75
# 22	19,14	28	**0,8937**	0,2393	80	19,14	28	**0,8937**	0,2393	80
# 23	21,33	29	**0,8469**	0,1693	126	9,03	28	**0,8972**	0,1599	119
# 24	30,44	28	**0,3427**	0,4989	61	17,93	26	**0,8782**	0,3384	53
# 25	25,55	27	**0,5435**	0,2388	107	24,82	28	**0,6377**	0,2533	98
# 26	22,88	28	**0,7389**	0,2724	84	22,17	28	**0,7735**	0,2879	77
# 27	28,16	26	**0,3506**	0,2874	98	21,68	24	**0,5982**	0,2492	87
# 28	10,73	13	**0,6333**	0,511	21	10,73	13	**0,6333**	0,511	21

Tabelle 7: Anpassungsergebnisse der negativen Binomialverteilung.

der teilweise fehlenden Normalverteilung der Variablen zu berechnende Spearman'sche Rangkorrelationskoeffizient zeigt ($\rho = .63$ bzw. $\rho = .62$ für die Parameter k und p, jeweils $p < 0.001$). Eine etwaige Abhängigkeit der Parameter von der (in der Anzahl der Sätze gemessenen) Textlänge ist dabei nicht erkennbar.

Allerdings ist in Bezug auf die Parameter k und p ein weiterer interessanter Zusammenhang zu beobachten, auf den in der Geschichte der Satzlängenforschung bislang noch nicht aufmerksam gemacht worden ist, und den es in Zukunft detailliert und systematisch zu verfolgen gilt. Dieser Zusammenhang betrifft eine auffällige und signifikante Abhängigkeit des Parameters p vom Parameter k: üblicherweise allerdings wird bei der Interpretation derartiger Zusammenhänge nicht der Parameter p herangezogen, sondern der sich als $1 - p$ berechnende Parameter q der negativen Binomialverteilung.

Abbildung 4 veranschaulicht den linearen Zusammenhang zwischen den Parametern

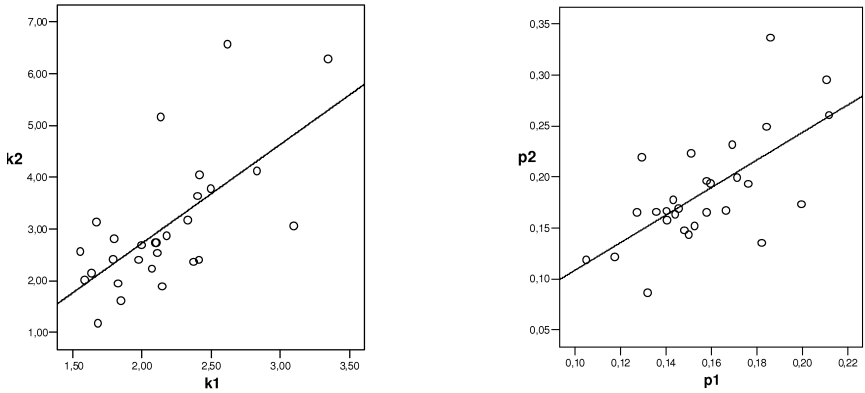


Abbildung 3: Parameter k und p der negativen Binomialverteilung für Satzdefinition 1 und 2.

q und k für beide Satzdefinitionen. Wie deutlich zu sehen ist, ist der Zusammenhang für Satzdefinition 2 wesentlich stärker; das bestätigt der aufgrund der teilweise fehlenden Normalverteilung zu berechnende Spearman'sche Rangkorrelationskoeffizient, der im Falle von Satzdefinition 1 einen Wert von $\rho = -.70$ aufweist ($p < 0.001$), im Falle von Satzdefinition 2 einen Wert von $\rho = -.91$ ($p < 0.001$).

Abgesehen davon, dass dieser Zusammenhang als ein starkes Argument für die Güte der Satzdefinition 2 angesehen werden kann, ergibt sich für die Theorie der Satzlängenforschung an dieser Stelle die Aufgabe, den Zusammenhang zwischen den beiden Parametern q und k der negativen Binomialverteilung einer qualitativen Interpretation auf der Grundlage umfangreicheren Datenmaterials zu unterziehen.

6 Resümee

Abschließend gilt es, die erhaltenen Resultate kurz zusammenzufassen:

1. Diskutiert wurde die bislang in der Satzlängenforschung vernachlässigte Fragestellung der Anwendung und Auswirkung unterschiedlicher Satzdefinitionen; diese Diskussion ist als zentral für jegliche weiterführende Untersuchungen zu sehen. Im vorliegenden Fall konnte gezeigt werden, dass die Anwendung von zwei (qualitativ prinzipiell als gleichwertig anzusehenden) Satzdefinitionen zu statistisch signifikanten Unterschieden des Mittelwertes führen kann. Als statistische Verfahren wurden dabei der (im gegebenen Fall die zu seiner Durchführung notwendigen Voraussetzungen nicht erfüllende) t-Test und der Mann-Whitney'sche U-Test angewandt. Es konnte gezeigt werden, dass nicht mehr als ca. 15% der Texte einen signifikanten Unterschied in den Mittelwerten aufweisen. Erklärbar ist dies einerseits durch die

	Satzdefinition 1			Satzdefinition 2		
Text	k_1	p_1	N	k_2	p_2	N
# 7	1,9746	0,1403	2938	2,4109	0,1582	2758
# 1	1,5810	0,1048	118	2,0227	0,1190	109
# 2	1,5507	0,1272	191	2,5731	0,1656	165
# 3	1,7865	0,1356	1493	2,4157	0,1660	1383
# 4	2,0949	0,1431	585	2,7412	0,1781	561
# 5	3,0963	0,1499	170	3,0636	0,1440	169
# 6	1,8216	0,1174	381	1,9531	0,1218	371
# 8	1,7964	0,1401	76	2,8140	0,1672	68
# 9	2,8281	0,2117	44	4,1330	0,2612	38
# 10	1,6654	0,1510	71	3,1466	0,2236	59
# 11	2,6144	0,1858	47	6,5782	0,3371	43
# 12	1,6329	0,1440	92	2,1535	0,1637	82
# 13	2,0681	0,1663	90	2,2424	0,1674	85
# 14	1,6774	0,1319	61	1,1899	0,0867	57
# 15	2,1435	0,1995	81	1,8966	0,1737	80
# 16	2,1066	0,1761	81	2,5422	0,1937	75
# 17	2,3279	0,1712	79	3,1800	0,1996	71
# 18	1,9943	0,1456	120	2,6939	0,1697	107
# 19	3,3410	0,2106	65	6,2973	0,2959	60
# 20	1,8440	0,1820	120	1,6238	0,1359	113
# 21	2,4949	0,1841	80	3,7872	0,2497	75
# 22	2,4091	0,1524	80	2,4091	0,1524	80
# 23	2,1748	0,1596	126	2,8750	0,1944	119
# 24	2,1318	0,1292	61	5,1725	0,2197	53
# 25	2,0984	0,1579	107	2,7400	0,1656	98
# 26	2,4008	0,1578	84	3,6458	0,1964	77
# 27	2,4113	0,1690	98	4,0521	0,2318	87
# 28	2,3706	0,1481	21	2,3706	0,1481	21

Tabelle 8: Die Parameter k und p der negativen Binomialverteilung.

Stabilität der beiden Satzdefinitionen, andererseits durch die gewählte homogene Textbasis (ausschließlich slowenische Prosatexte). Beide angewandten Tests zeigen insbesondere auf der Ebene des Gesamtkorpus und bei dem längsten Text signifikante Unterschiede, was den Schluss nahelegt, dass signifikante Unterschiede erst bei einer größeren Anzahl von Satzlängen mit unterschiedlicher Länge zum Tragen kommen.

2. Aufgrund der eingeschränkten Aussagekraft von Mittelwerten (die aber dennoch eine in der Satzlängenforschung sehr beliebte Kenngröße sind) wurden auch die Profile der sich aufgrund der Satzdefinitionen ergebenden Satzlängenverteilungen einem statistischen Signifikanztest in Form eines Vergleich von Kurtosis und Schiefe unterworfen. Es zeigt sich hierbei, dass für keinen der Texte ein signifikanter Unterschied nachgewiesen werden kann.

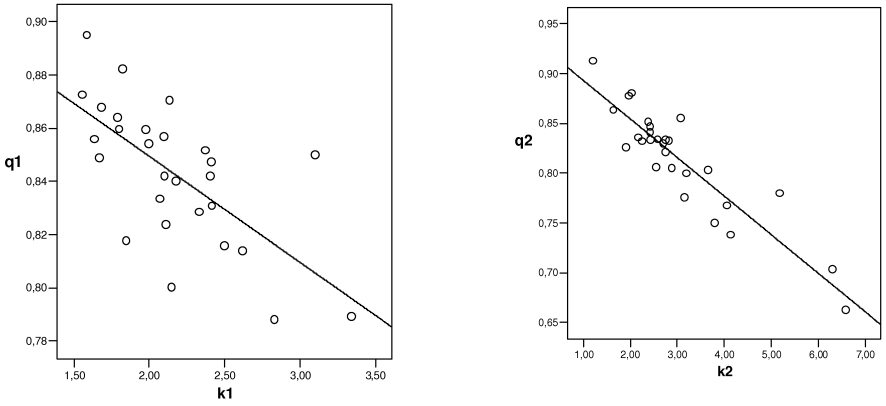


Abbildung 4: Abhängigkeit des Parameters q vom Parameters k der negativen Binomialverteilung für Satzdefinition 1 und 2.

Insofern würde man bereits a priori feststellen können, dass auch die Anwendung von unterschiedlichen Satzdefinitionen auf der Ebene der theoretischen Modellierung keinen bzw. nur geringen Einfluss haben kann. Dieser Befund konnte durch entsprechende empirische Überprüfungen bestätigt werden, in denen die negative Binomialverteilung als geeignetes theoretisches Modell der Satzlängenverteilung in allen zugrunde gelegten slowenischen Prosatexten nachgewiesen werden konnte. Darüber hinaus lassen sich auf der Ebene der theoretischen Modellierung folgende Resultate anführen:

1. Die beiden Satzdefinitionen lassen einen Zusammenhang erkennen, der sich an der negativen Korrelation der jeweiligen aus den theoretischen Verteilungen resultierenden Parameterwerte (k_1 und k_1 bzw. p_1 und p_2) nachweisen lässt. Insofern lässt sich postulieren, dass die Anwendung der beiden angewandten Satzdefinitionen zu einer *systematischen Verschiebung der Parameter* führt.

2. Als wichtig und richtungsweisend anzusehen ist auch der Befund eines statistischen Zusammenhangs zwischen den beiden Parametern k und p der negativen Binomialverteilung. Die beiden Parameter sind bei beiden Satzdefinitionen hoch korreliert, wobei der deutlich ausgeprägtere Zusammenhang bei Satzdefinition 2 ($\rho = -.91$) als ein Argument dafür zu interpretieren ist, dass die Satzdefinition 2 als die qualitativ adäquatere Satzdefinition anzusehen ist.

Insgesamt ist dieser Beitrag als eine zentrale Detailuntersuchung der Satzlängenforschung zu sehen, wobei eine Ausweitung auf weitere Textsorten und Sprachen wünschenswert wäre (vgl. Kelih, 2002). Darüber hinaus kann mit der vorgestellten Satzdefinition systematisch die Frage untersucht werden, inwiefern die Satzlänge und daraus be-

rechnete Kenngrößen ein adäquates Mittel für eine Klassifizierung von Texten, Textsorten bzw. Funktionalstilen herangezogen werden kann (vgl. Kelih et al., 2006).

Literatur

Altmann, G. (1988a). Verteilungen der Satzlängen. In K. P. Schulz (Hrsg.), *Glottometrika 9*, S. 147–161. Bochum: Brockmeyer.

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

Altmann, G. (1992). Das Problem der Datenhomogenität. In B. Rieger (Hrsg.), *Glottometrika 9*, S. 287–298. Bochum: Brockmeyer.

Altmann, G. und W. Lehfeldt (1980). *Einführung in die Quantitative Phonologie*. Bochum: Brockmeyer.

Antić, G., E. Kelih und P. Grzybek (2006). *Contributions to the Science of Language. Word Length Studies and Related Issues*, Kapitel *Zero-syllable Words in Determining Word Length*, S. 117–156. Dordrecht, NL: Springer.

Bolton, H. C. und A. Roberts (1995). On the comparison of literary and scientific styles: The letters and articles of Max Born, FRS. *Notes and Records of the Royal Society of London 49*(2), 295–302.

Bünting, K. D. und H. Bergenholtz (1995). *Einführung in die Syntax*. Stuttgart: Beltz.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology 32*, 221–233.

Grinbaum, O. N. (1996). *Prikladnoe jazykoznanie*, Kapitel *Komp'juternye aspekty stilemetrii*, S. 451–463. Sankt-Peterburg: Izd. S.-Peterburgskogo universiteta.

Grotjahn, R. und G. Altmann (1993). *Contributions to Quantitative Linguistics*, Kapitel *Modelling the Distribution of Word Length: Some Methodological Problems*, S. 141–153. Dordrecht: Kluwer Academic.

Grzybek, P. (2000). *Slovo vo vremeni i prostranstve. K 60-letiju professora V. M. Mokienko*, Kapitel *Zum Status der Untersuchung von Satzlängen in der Sprichwortforschung – Methodologische Vor-Bemerkungen*, S. 430–457. Moskva: Folio-Press.

Holmes, D. I. (1994). Authorship attribution. *Computers and the Humanities 28*, 87–106.

Janson, T. (1964). The problems of measuring sentence-length in classical texts. *Studia Linguistica 18*, 26–36.

Karlgren, J. und D. Cutting (1994). Recognizing text genres with simple metrics using discriminant analysis. In M. Nagao (Hrsg.), *Proceedings of COLING 94*, S. 1071–1075.

Kelih, E. (2002). Untersuchungen zur Satzlänge in russischen und slowenischen Prosatexten. Band 1 und 2. Diplomarbeit, Karl-Franzens-Universität Graz, Institut für Slavistik, Graz.

Kelih, E. und P. Grzybek (2004). Häufigkeiten von Satzlängen: Zum Faktor der Intervallgröße als Einflussvariable (am Beispiel slowenischer Texte). *Glottometrics 8*, 23–41.

Kelih, E., P. Grzybek, E. Stadlober und G. Antić (2006). *Text Classification: The Impact of Sentence Length*. Heidelberg: Springer.

Kjetsaa, G. (1984). *The Authorship of the Quiet Don*. Oslo: Solum Forl.

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

Lewis, P. A. und E. J. Orav (1989). *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*. Pacific Grove, CA: Wadsworth & Brooks.

Mistrík, J. (1973). Eine exakte Typologie von Texten. In *Arbeiten und Texte zur Slavistik*, Band 3. München: Verlag Otto Sagner.

Niehaus, B. (1997). Untersuchung zur Satzlängenhäufigkeit. In K.-H. Best (Hrsg.), *Glottometrika 16*, S. 213–276. Trier: WVT.

Orlov, J. K. (1982). Linguostatistik: Aufstellung von Sprachnormen oder Analyse des Redeprozesses? (Die Antinomie „Sprache-Rede" in der statistischen Linguistik). In J. K. Orlov, M. G. Boroda und I. Š. Nadarejšvili (Hrsg.), *Sprache, Text, Kunst. Quantitative Analysen*, S. 1–55. Bochum: Brockmeyer.

Pieper, U. (1979). *Über die Aussagekraft statistischer Methoden für die linguistische Stilanalyse*. Tübingen: Narr.

Pravopis (1990). *Slovenski pravopis. I: Pravila*. Ljubljana: Državna založba Slovenije.

Sherman, L. A. (1888). Some observations upon the sentence-length in English prose. In *Studies of the University of Nebraska*, Band 1, S. 119–130. University of Nebraska.

Sichel, H. S. (1974). On a distribution representing sentence-length in written prose. *Journal of the Royal Statistical Society (A) 137*, 25–34.

Sigurd, B., M. Eeg-Olofsson und J. van de Weijer (2004). Word length, sentence length and frequency – Zipf revisited. *Studia Linguistica 58*(1), 37–52.

Smith, M. W. A. (1983). Recent experience and new developments of methods for the determination of authorship. *Bulletin of the Association for Literary and Linguistic Computing 11*(3), 73–82.

Teigeler, P. (1968). *Verständlichkeit und Wirksamkeit von Sprache und Text*. Stuttgart: Verlag Nadolski.

Tuldava, J. (1993). Measuring text difficulty. In G. Altmann (Hrsg.), *Glottometrika 14*, S. 69–81. Bochum: Brockmeyer.

Wake, W. C. (1957). Sentence-length distributions of Greek authors. *Journal of the Royal Statistical Society 120*, 331–346.

Weiß, H. (1968). *Statistische Untersuchungen über Satzlänge und Satzgliederung als autorenspezifische Stilmerkmale. (Beitrag zur mathematischen Analyse der Formalstruktur von Texten)*. Disseration, TH Aachen.

Williams, C. B. (1940). A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika 31*, 356–361.

Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose. *Biometrika 30*, 363–390.

Software

Altmann-Fitter (2000): *Iterative Fitting of Probability Distributions*. Lüdenscheid: RAM Verlag.

Georg Rehm

Language-Independent Text Parsing of Arbitrary HTML-Documents. Towards A Foundation For Web Genre Identification

This article describes an approach to parsing and processing arbitrary web pages in order to detect macrostructural objects such as headlines, explicitly- and implicitly-marked lists, and text blocks of different types. The text parser analyses a document by means of several processing stages and inserts the analysis results directly into the DOM tree in the form of XML elements and attributes, so that both the original HTML structure, and the determined macrostructure are available at the same time for secondary processing steps. This text parser is being developed for a novel kind of search engine that aims to classify web pages into web genres so that the search engine user will be able to specify one or more keywords, as well as one or more web genres of the documents to be found.

1 Introduction

This article describes an approach to language-independent text parsing of arbitrary HTML documents. Taking only the HTML element tree and the "visual semantics" (i. e., the canonic way in which a browser renders a certain HTML construct) of elements and attributes into account, we are able to determine macrostructural properties (and in some cases even author-intended functions) of subtrees that will be helpful in several applications that involve processing natural language, especially within the contexts of automatic web genre identification and information extraction.

The main purpose of the text parser described in this article is to abstract from the often highly complex HTML element tree: instead of having to deal with trees containing one node for each HTML element, we want to treat web pages as trees containing only one node for each paragraph, headline, list structure etc. For this purpose, the text parser analyses a document by means of several recursive DOM-based methods that utilize, amongst others, XPath expressions in order to compute features for each subtree. Along with the concrete HTML elements and attributes of a certain subtree, these features can be used to determine its textual function. As soon as one such function is found, the respective subtree is encapsulated by means of a special XML element, the namespace of which is introduced immediately after the initial transformation of arbitrary HTML into well-formed XHTML code. This additional analysis namespace enables a novel view of an HTML document – a view that does not necessarily include every HTML element, but only the macrostructural information that we need in order to be able to tackle the abovementioned processing stages. At the same time, the original HTML element tree is still available in its own namespace for analysis refinement and secondary processing steps.

One of the reasons why it is necessary to analyse HTML documents in the manner proposed in this paper is the phenomenon known as "tag abuse" (Barnard et al., 1996): from a text-structural and text-functional point of view, macrostructural components contained in HTML documents can be *identical* but they can be realized by means of fundamentally *different* HTML elements. It is necessary to reconstruct the macrostructural building blocks intended by the author of a document, as these building blocks cannot be inferred from their respective HTML elements that might be affected by tag abuse (Eiron and McCurley, 2003). This process of reconstructing a document's macrostructure brings us back to the visual semantics of the corresponding HTML elements and attributes. Especially their typographical features need to be examined in order to determine the intended function of a building block. HTML's vocabulary is rather limited, but, ideally, information about navigation bars, headlines, and paragraphs of running text should be available in a consistent set of automatically annotated XML elements.

The remainder of this paper is structured as follows: Section 2 sketches related work on parsing HTML documents. Section 3 introduces the broader context of this article, namely a novel theoretical framework for the automatic identification of web genres. Section 4 describes the text parser, its processing stages, the graphical web front end, and two visualization methods.

2 Related Work

There is a large number of approaches to parsing and processing web pages, driven by objectives as different as the improvement of search and navigation facilities, wrapping, information extraction, automatic summarization, or the adaptation of HTML pages for mobile devices. Carchiolo et al. (2003) parse HTML element trees based on "primary tags" which constitute "primary nodes" (such as `table`, `p`, `map`, `hr`, and `a`). After computing weighted features for each primary node (relative and absolute depth, number of leaves etc.), "collections" are built which are finally mapped onto "logical sections" ("document", "header", "footer", "index", "logical data", and "interactive" sections) defined as "parts of a page each of which collects related information". Yang et al. (2003) present a DOM-based bottom-up approach in order to detect "semantically meaningful clusters", i. e., the implicit schema in template-driven HTML pages. WordNet is employed to find related concepts that can be used as constraints in the structural analysis. Finally, individual clusters are labeled using both statistical and symbolic methods. Chan and Yu (1999) sketch an approach for extracting web design knowledge. First, a "canonical form" is produced based on "primitive" (`br`, `b`, `i`, `font`, `table` etc.) and "compound tags" (`h1`–`h6`, `p`, `ol` etc.). Secondly, "objects" are identified, based on "object tags" and the word count at the leaf level. Three "design knowledge bases" are constructed containing "site layout and navigation", "web page objects", and "web page layouts". Chen et al. (2001) describe the function-based object model (FOM) which was developed for the automatic conversion of HTML documents to WML (Wireless Markup Language). For this process, redundant objects need to be removed and the content needs to be condensed due to the display constraints of mobile devices. With regard to several features of a web page,

Chen et al. (2001) apply statistical methods, decision trees, and visual similarity pattern detection (cf. Song et al., 2004) based on an internal rendering of a page, to determine the specific FOM of individual objects which fall into categories such as information object, navigation object, interaction object, or decoration object. Finally, the non-redundant objects can be used to create a WML-version of the original HTML document.

Myllymaki (2001) describes an information extraction system which is based upon the simple, but powerful notion of processing legacy HTML documents with XML tools. For this purpose, the HTML documents are automatically converted to well-formed XHTML markup, which is, of course, based upon XML, therefore enabling the use of arbitrary XML tools and standards. Myllymaki's ultimate goal is to replicate the databases behind DB-driven web sites. Several key aspects of this wrapping approach can be efficiently carried out by means of cascaded XSLT stylesheets that need to be manually finetuned. Chung et al. (2001, 2002) convert "topic specific" web pages into XML-formats using a DOM-based approach. Examples of topics are "product descriptions", "bibliographies", or "resumés". In the "concept identification" stage, the content of a document is tokenized and matched onto manually predefined concepts. In the subsequent "tree restructuring" phase, the nodes of the intermediate tree structure are rearranged so that "the resulting tree reflects the logical layout of the information carrying objects". In this approach, HTML nodes are gradually replaced by XML element nodes (such as `degree` or `thesis` for the "resume" topic) using the DOM engine. Gupta et al. (2003) present a technique for detecting the content in arbitrary web pages using several filtering stages that process a document's DOM tree recursively, removing and modifying specific nodes, "leaving only the content behind". For example, using the "advertisement remover", certain embedded images whose URLs point to blacklisted web servers can be deleted. By means of the "link list remover", certain link lists contained in table cells can be removed. Gupta et al. (2003) list several potential fields of application for their tool: adaptation of HTML pages for mobile devices, speech rendering for the visually impaired, and summarization. Chen et al. (2003) describe a similar technique with the specific goal of adapting web pages which use the widely employed three-column-layout, to small form-factor displays. The aim of the analysis is to "recover the content structure based on the clues the author embeds in a web page" so that a single web page can be split up into multiple documents that can, in turn, be efficiently browsed on mobile devices. In the DOM-tree based page analysis stage, the "semantic structure" is identified by classifying each node into one of the "content block" categories "header", "footer", "left side bar", "right side bar", and "body". Furthermore, explicit (tags such as `hr`, `table`, `td` and `div`) and implicit boundaries ("blank areas created intentionally") are detected. Afterwards, a document can be split up into multiple documents during the page adaptation stage. Buyukkokten et al. (2001) developed the Power Browser that is able to automatically summarize web pages using an "accordion" metaphor. First, the tool identifies "semantic textual units" within a page and rearranges them into a tree structure that is constructed using the properties of the individual units.

HTML's `table` tag is one of the main instruments in web design. Therefore, several approaches focus upon the automatic processing of tables. Hurst (2002) and Cohen et al.

(2002) try to distinguish genuine tables from tables which are only used to achieve a specific layout, by means of supervised machine learning methods. The features are partially based on a DOM representation and comprise, amongst others, "single HTML row", "single HTML column", and "bag of tags". The authors report a recall of 92–96% and a precision of 95–100% for an evaluation with 89 positive and 250 negative examples. Penn et al. (2001) and Wang and Hu (2002) describe alternative approaches for the detection of genuine tables. Alam et al. (2003) identify tables which are used to mimic lists. Lim and Ng (1999) try to reconstruct the "intended hierarchy of the data content of the table".

3 Genres in the World Wide Web

The Hypnotic project (*Hypertexts and their organisation into a taxonomy by means of intelligent classification*) aims to develop methods and concepts of identifying the respective web genre of arbitrary web pages with the ultimate goal of providing a web genre-enabled search engine (Rehm, 2005). This search engine will ultimately offer a traditional keyword-based search interface augmented with a filtering layer that will give the user the ability to specify the web genre(s) of the documents he or she wants to find (Rehm, 2002). In addition, we see web genre classification as a very promising preprocessing step for novel information extraction tasks (Rehm, 2004a,b).

3.1 Web Genres and Web Genre Modules

The term web genre is a novel concept based upon the classic text linguistics notion of text genre (also known as text type, or text sort). Prominent and extensively researched examples are: *Weather Report*, *Cooking Recipe*, *Letter*, and *Scientific Article*. Some text genres even contain subgeneric variants: A *Love Letter* and a *Business Letter* share a certain number of features that are responsible for the fact that these text genres are both instances of the more general *Letter*, but differ with regard to other features, justifying the assumption that we are dealing with two distinct text genres. Furthermore, text genres can be grouped into multiple hierarchies, e. g., text types that people send to one another (*Letter*, *Fax*, *Text Message*, *Invoice* etc.), or text types in use within academic communities (*Article*, *Memo*, *Technical Report*, *Final Report of a Project*, *Master's Thesis*, or *Ph. D. Thesis*). Traditionally, and on the most general level, text genres can be characterised by the three features *content*, *form*, and *function*. The set of text genres in existence is by no means fixed, but is gradually and unconsciously extended as soon as, for example, a new technology emerges. Mobile phones have introduced a new means of communication, the text message (or SMS, short message service) now being used more than a billion times each day throughout the world. Due to the constraint that one text message must not exceed 160 characters, a novel, heavily abbreviated, telegram-like style of writing has established itself amongst those who use this medium, often borrowing and even re-inventing acronyms and abbreviations that have been reported in computer-mediated communication (CMC) studies on linguistic phenomena within email, Usenet and IRC in

the last ten years (see, e. g., Haase et al., 1997). We do not want to discuss whether text messages can be justified as a new text genre but would like to point out the fact that a newly introduced medium often (and most of the time, inevitably) leads to the emergence of novel text genres, based on new and subsequently recurring communicative situations.

One such area in which new genres develop rapidly, is the World Wide Web. We even go as far as to give this new class of text genres a label of its own: Web genres. World-wide, hundreds of millions of people use the web each day, searching for, reading, and examining web pages. Most of the time, they find the desired information, but, sometimes, the expectations of a user are not met and leave him wondering why certain information is not contained within a specific web page or web site. For example, trying to locate the volume and issue of a certain scientific article needed for a list of references, a user might find the personal home page of the author but a list of his publications may not be contained in his or her web site (cf. Dillon and Gushrowski, 2000). Unfulfilled expectations like these can be explained in terms of the web genre concept: Since the beginning of the 1990s, millions of people have written or developed personal home pages and based the content and layout of their own web pages on the HTML documents they previously encountered. In the early 1990s, these pages were a direct reflection of the possibilities this technological breakthrough offered (Furuta and Marshall, 1996), a distributed hypertext system that allowed people to show off photos of their children, a trip to the Caribbean, or lists of their favourite records. Over the years, the medium came of age, more and more people used the web and, to continue this example, especially the authors of personal home pages intuitively realized that, somehow, certain information does in fact 'belong' on the web while other information does not.

If we consider the *Academic's Personal Home Page*, a subgeneric variant of the *Personal Home Page* (Rehm, 2002), a large variety of information could be found on academics' home pages in the mid-90s, but nowadays there is, as far as content, form, and function are concerned, an almost uniform picture: On their home pages, academics first give their name (often in a large font and accompanied by a photo) and function within the university they work in, they list contact information, current research projects, publications etc. Another web genre is the *Entry Point of a Department* in which, in most cases, lists of staff members, publications, research projects, directions and several other informational units can be found.[1]

Hyperlinks, the most fundamental concept of hypertext, break the boundaries in terms of what constitues a specific web genre. A certain informational unit often contained in a *Personal Home Page* comprises contact information (email address, street address, phone and fax numbers etc.). If this informational unit is embedded within a page, it has a status very different from cases where this information is contained in a file of its own linked to from the home page itself (cf. Mehler et al., 2004). In other words, web genres are by no means monolithic entities, but rather highly modular concepts (see Haas and Grams, 2000): some web genres can only act as web genre instances on their own (e. g., *Personal Home Page*) while others — e. g., the abovementioned *Contact Information* —

[1]Note that, in our theoretical framework, the categories labeled "resume", "product description" etc. by Chung et al. (2001, 2002) are not "topics" but web genres.

can, in addition, be re-used within more general web genres like *Personal Home Page* or *Institutional Home Page*. For this reason, one of our main assumptions is that web genres are composed of web genre modules. For a specific web genre, web genre modules can be either compulsory (e. g., the *List of Publications* in the abovementioned example), or optional (e. g, *Photo Gallery* within *Academic's Personal Home Page*). Furthermore, each web genre has a default assignment with regard to the <content, form, function> triple that is preset by the compulsory modules involved, but which can be modified by the presence of optional modules. Additional details on the relationship between web genres and web genre modules can be found in Rehm (2002), a paper that, furthermore, reviews previous studies with regard to this line of research. As of yet, there are no definitive reports on either the number of web genres or on their hierarchical or taxonomic structure: In most of the related work (see, e. g., Crowston and Williams, 1997, Shepherd and Watters, 1999), random samples of web pages have been drawn using the "surprise" link offered by search engines in the late-90s, i. e., no constraints whatsoever have been employed with regard to the sampling, resulting in very coarse and extremely general lists of dozens of contrasting web genres which are mostly unrelated to one another.

3.2 A Corpus-Database of Web Pages

Due to the abovementioned methodological problems of previous approaches to identifying a coherent set of web genres, we decided to concentrate exclusively on academic web pages: we crawled Germany's academic web (in 2001 and 2002) in order to build a static, and therefore stable corpus of this clearly defined domain, which is perfectly suited for web genre research, without being forced to tackle *all* web genres in existence.

Of the circa 260 German universities (incl. general, technical, specialized and private universities, as well as polytechnics), the corpus contains snapshots of 100 universities. The most important category for our project is "general universities" comprising local mirrors of all the HTTP servers of Germany's 62 'traditional' universities. In total, our crawler traversed 14,968 web servers. Of the 16.2 mio. files visited by the crawler, 4.3 mio. were mirrored in the local filesystem. We further limited the crawl to only those HTML documents written in German, by employing a lexicon-based language identification tool (Rehm, 2001). Table 1 shows the contents of our collection. It can be seen that 3.9 mio. (46%) of the web pages available in Germany's academic web are written in German. Furthermore, a web-accessible PHP- and MySQL-based corpus-database has been developed to facilitate web server and document access and the random generation of document samples. The sample generation can be finetuned with about 20 different parameters, restricting the documents to be randomly included in a sample to certain hostname or filename patterns, domains, file sizes etc. The key components of the corpus-database are an Apache web server which delivers the locally mirrored HTML documents and a MySQL database that contains metadata about these documents, so that efficient retrieval and access methods can be provided. The contents and structure of the HTTP response header (Fielding et al., 1999) acted as a blueprint for the metadata tables (Rehm, 2001) which have not only been populated with the HTTP

Number of universities:	100
• General universities (*complete*)	62
• Technical universities (*complete*)	12
• Music and arts universities (*partial*)	5
• Business universities (*partial*)	5
• Misc. (*partial*)	16
Traversed web servers:	14,968
Web servers operating on port 80:	13,885
Files available via HTTP:	16,196,511
Number of HTML documents:	8,465,105
Total size of all web servers:	701,464.29 MB
Total size of the corpus:	40,914.99 MB
Running word forms (total; `text/html` only):	1,138,794,715
Running word forms (unique; `text/html` only):	12,120,162
Total number of files in the corpus-database:	4,294,417
• Media type `text/html`:	3,956,692
• Media type `text/plain`:	270,400
• Media type `text/css`:	35,651
• Media type `text/xml`:	25,871
• Media type `text/sgml`:	956
• Media type `message/news`:	490
• Media type `message/rfc` :	436

Table 1: Contents of the corpus-database.

response header information of the documents stored on the database-server, but also with the response header data of *all* 16.2 mio. documents and files that were referenced in the original web pages.

3.3 The Document Sample Testbed

The analysis of individual document samples is our primary tool for the identification of specific web genres. After the generation of a sample, the corresponding data set is imported into the corpus-database for future access. Although the sample analysis has to be carried out manually, the database supports the analyser by providing an HTML form of the features to be examined in a pop-up window. After the form for a certain document has been filled out, the information can be saved in the database and later visualized with the analysis data of the other documents in a spread-sheet-like manner.

Several samples have been analysed with the ultimate goal of devising a coherent web genre taxonomy for documents originating within the domain of academia. The examination of an initial sample of 200 web pages was carried out in order to get an initial impression of this domain (Rehm, 2002). Another sample which has been analysed is made up of 727 web pages containing the entry pages of the first 35 universities and the first level of pages linked to from the 35 entry pages. The goal of this analysis was to define the upper structure of the web genre taxonomy (Rehm, 2004b). A third, randomly selected sample of 750 documents has been analysed in order to finalize the leaf level of the taxonomy, which has been modelled in OWL (Rehm, 2005).

Additional samples each contain web pages of specific web genres. One of these samples is used as a testbed for the development phase of the text parser described in the next section, comprises 100 documents of the web genre *Academic's Personal Home Page* and has been collected semi-automatically from web servers offering personal home pages. A list of those pages was randomly shuffled and 100 documents were put into the sample if the web page was (a) the personal home page of an academic, (b) written in German, (c) belonged to one single person (in contrast to, e. g., a research group), (d) primarily dealt with the job of the author at the university and (e) did not use framesets. Table 1 illustrates the (abbreviated) results of this analysis, the primary goals of which were to create a list of the web genre modules involved in this web genre and to determine the status of each module (compulsory vs. optional) based on their individual frequencies within this sample (threshold: 50). The table introduces several new concepts. Web genre modules can be either atomic or complex entities: complex modules consist of two or more features which have to be present in order to instantiate the respective module. Atomic modules consist of only one certain feature. The *Academic's Personal Home Page* is a subgeneric variant of *Personal Home Page*. Therefore, each module in the table can be either general (e. g., *Contact information*) or specific (e. g., *Office hours*) with regard to the more general *Personal Home Page*. Based on an analysis such as the one shown in table 1, we can deduce a definition of the respective web genre in order to specify its content, form and function. Our definition of the *Academic's Personal Home Page*, based on de Saint-Georges (1998), is contained in Rehm (2002).

4 The Text Parser

The previous section introduced our theoretical framework and briefly sketched our ultimate goal of building a web genre-enabled search engine. In order to automatically detect and identify different web genres, we need to take both the content and the structure of a web page into account so that we are able to compute features which can, in turn, be fed to a classification algorithm. A second goal could almost be considered a by-product if the identification of web genres proves to be feasible: If we are able to detect the web genre of a given web page, we could develop means of instantiating the web genre modules involved. Hence, we could extract informational units on a fine-grained level such as the one presented in table 1, in an automatic way. In other words, if we know that a certain HTML document belongs to a certain web genre, we know what kind of content to expect. Based on these expectations, we could try to tackle the information extraction problem (Cowie and Wilks, 2000) on a novel, web genre module-based level.

We developed a text parser for arbitrary web pages in order to lay a foundation for computing features for the web genre classifier, and for the information extraction task. This prototype, implemented in Perl, is embedded in the corpus-database and several means of visualization of the analysis results can be accessed in the "document view" mode by means of the web front end. We developed the parser based on the 100 documents of the *Academic's Personal Home Page* contained in the sample described in section 3.3. The parser's design is based on several principles:

Level	Description	Module/Feature	Status	Number
Atomic	**Explicit greeting**	general	optional	14
Complex	**Identification**	general	compulsory	—
Feature	*Name of the owner of the page*	general	compulsory	100
Feature	*. . . accompanied by an academic title*	specific	compulsory	69
Feature	*. . . accompanied by a job title*	general	optional	27
Feature	*. . . accompanied by an affiliation*	general	optional	34
Feature	*. . . accompanied by a picture of the author*	general	compulsory	54
Complex	**Independent affiliation**	general	compulsory	—
Feature	*Name of the university (in machine readable form)*	general	compulsory	75
Feature	*Logo graphic of the university*	general	optional	16
Atomic	**Alternate version in different language**	general	optional	75
Complex	**Contact information**	general	compulsory	—
Feature	*Street address (university, street, zip, city etc.)*	general	compulsory	90
Feature	*Explicit postal address*	general	optional	8
Feature	*Phone number*	general	compulsory	86
Feature	*Phone number (secretary)*	general	optional	7
Feature	*Fax number*	general	compulsory	66
Feature	*Email address*	general	compulsory	98
Feature	*URL of this home page*	general	optional	4
Feature	*Room/office number*	general	optional	30
Feature	*Send SMS text message*	general	optional	1
Feature	*PGP public key or PGP fingerprint*	general	optional	2
Feature	*X.500 entry*	general	optional	2
Feature	*Directions*	general	optional	2
Feature	*Office hours*	specific	optional	25
Feature	*Address (private)*	general	optional	18
Feature	*Phone number (private)*	general	optional	22
Feature	*Mobile phone number (private)*	general	optional	3
Feature	*Fax number (private)*	general	optional	7
Feature	*Email address (private)*	general	optional	5
Feature	*URL of the private home page*	general	optional	2
Complex	**Contact information (secretary)**	specific	optional	—
Feature	*Name*	general	optional	8
Feature	*Street address*	general	optional	3
Feature	*Room/office number*	general	optional	4
Feature	*Opening hours*	general	optional	5
Feature	*Phone number*	general	optional	6
Feature	*Fax number*	general	optional	6
Feature	*Email address*	general	optional	6
Complex	**Contact information (staff members)**	specific	optional	—
Feature	*Name*	general	optional	7
Feature	*Listing of multiple entries*	meta	optional	6
Feature	*Address*	general	optional	2
Feature	*Room/office number*	general	optional	3
Feature	*Phone number*	general	optional	4
Feature	*Email address*	general	optional	4
Feature	*Names of student assistants*	general	optional	2
Complex	**Academic Profile**	specific	optional	—
Feature	*List of courses*	specific	optional	49
Feature	*Position(s) within the university*	specific	optional	7
Feature	*General student information*	specific	optional	3
Feature	*Suggested titles of final theses*	specific	optional	2
Complex	**Scientific profile**	specific	compulsory	—
Feature	*List of publications*	specific	compulsory	71
Feature	*Research interests*	specific	compulsory	50
Feature	*Research projects*	specific	optional	22
Feature	*Prominently displayed books and/or journals*	specific	optional	6
Feature	*List of talks or presentations*	specific	optional	5
Feature	*Membership in professional associations*	specific	optional	4
Feature	*Technology transfer*	specific	optional	1
Atomic	**C. V.**	general	compulsory	60
Atomic	**Interesting links**	general	optional	12
Complex	**Relevant links**	general	optional	—
Feature	*Link to one's home department/school/insitute*	specific	optional	49
Feature	*Link to one's home university home page*	specific	optional	36
Feature	*Link to one's home faculty home page*	specific	optional	23
Atomic	**Most recent update**	universal	optional	42
Atomic	**Access counter**	universal	optional	11
Atomic	**Guestbook**	universal	optional	1

Table 2: The Web genre *Academic's Personal Home Page*.

Simplicity and robustness We want to keep the algorithms and methods as simple, robust, and general as possible, so that they are, in theory, applicable to all web pages in existence.

Domain- and language-independence We do not want to restrict the functionality of the text parser in any way, i. e., we do not intend to process web pages of a specific domain or language only.

Non-destructive inline analysis annotation We want to keep our analysis results directly within the source document, i. e., we want to keep the original (X)HTML instance and augment this structure with elements and attributes of the analysis namespace (`hypnotic:`) step by step. This principle guarantees that we can access the original data at any point.

Maximum detection of implicit structure We would like to detect as much hidden structure as possible. For example, the authors of web pages tend not to fully exploit HTML's vocabulary, i. e., if an author wants to create a headline, HTML provides the headline-tags `h1` to `h6`, but most authors do not think in terms of tags (logical or structural markup) but rather in terms of font sizes, hence they use physical markup, i. e., the `font` element along with its attribute `size` and a relative or absolute value. Explicit headlines can be detected easily, but, in addition, we also want to find headlines (and sub-headlines) realized by means of the `font` element.

Use of XML standards wherever possible This principle says that we want to use and exploit current XML standards and techniques as much as possible. For example, the Document Object Model (DOM, Hors et al., 2000), XPath (Clark and DeRose, 1999) and XSLT (Clark, 1999) offer several key advantages which enable us to concentrate on the analysis algorithm (DOM, XPath) and the visualization of the analysis results (XSLT).

4.1 Converting HTML into XHTML Markup

The processing of arbitrary HTML documents (which potentially contain markup errors, unknown tags etc.) using XML techniques requires an initial conversion into at least well-formed XHTML code (Myllymaki, 2001). For this HTML to XHTML conversion, a Perl module has been developed which encapsulates the two packages tidy (`http://tidy.sourceforge.net`) and HTML::TreeBuilder (available in CPAN). Using a specific set of configuration parameters, tidy is able to output well-formed XHTML code. Both packages implement rules to cope with invalid HTML structures.

As tidy's rules to correct invalid HTML code are more robust than HTML::TreeBuilder's, our Perl module first tries to pass the source web page through tidy in order to create an XHTML version. If this does not work, we use HTML::TreeBuilder as a fallback tool to repair the HTML input, which is then passed on to tidy again. We evaluated this method by running 10,000 randomly selected documents from the corpus through the module: 98.7% of all documents were successfully transformed to XHTML, the

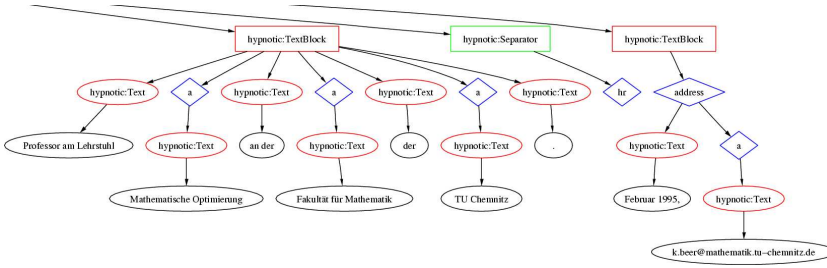


Figure 1: An excerpt from a DOM tree after the automatic insertion of two `hypnotic:TextBlock` elements.

fallback component HTML::TreeBuilder had to be activated 270 times. In addition, we processed these XHTML document instances using the non-validating parser expat (employing XML::Parser). Only 5 of the 9,872 successfully converted documents raised error messages, most of which were caused by character encoding errors. Finally we tried to parse the resulting documents against the XHTML 1.0 Transitional DTD by means of the XML/SGML parser onsgmls. Most of the error messages we found did not relate to malformed markup structures but were rather caused by incorrect attribute values (extremely often in `valign`), or unknown elements (e. g., `blink` or `spacer`).

4.2 The Parsing Algorithm

The parsing algorithm operates in several cascaded stages with increasing complexity and is based on the formal specification of the Hypertext Markup Language as defined in HTML 4.01 (Raggett et al., 1999). The three most important features for the algorithm are those HTML elements which cause a new paragraph to be created while the document is rendered within a browser, a comprehensive analysis of the current font size at any point within the DOM tree, and a rudimentary image analysis. As mentioned earlier, the main goal of this algorithm is to introduce as much explicit structure as possible: we would like to be able to abstract from the often highly complex HTML tree and instead, treat an arbitrary document on a more coarse-grained level.

Stage 1: Basic Annotation of Nodes

This initial stage preprocesses the DOM tree recursively and introduces analysed information as new attributes of our `hypnotic:` namespace within the original XHTML elements.

Each element within the tree receives an attribute `Path` containing an absolute XPath-like path specification (e. g., `/xhtml:html[1]/xhtml:body[1]/xhtml:h1[2]`). Furthermore, links and words (defined as any kind of whitespace-separated token) are counted on both the local and the subtree level. The font analysis computes the current font size for each element, relative to the base font size which can be preset using HTML's `basefont`

```
@s: The sequence of hypnotic:Text elements
@b: Paragraph elements: <p>, <ul>, <ol>, <dl>, <td>, <blockquote>, <div>
$n: The current element
$p: The previous element

while (next element) {
    if ($n == "<hypnotic:Icon>") {
        push(@s, $n);
    }
    if ($n == "<hypnotic:Separator>") {
        markTextBlocks(@s);
    }
    if (($n == "<br>") && ($p == "<br>")) {
        markTextBlocks(@s);
    }
    if ($n is element of @b) {
        markTextBlocks(@s);
    }
    if ($n is not contained in an 'open', partial tree, dominated by a
      node contained in @b) {
        markTextBlocks(@s);
    }
    if ($n == "<hypnotic:Text>") {
        if (!($n->FontSize == $p->FontSize +/-  )) {
            markTextBlocks(@s);
        }
        push(@s, $n);
    }
}
```

Listing 1: `findTextBlocks($r)` in pseudo-Perl code.

element. The base font size is mapped onto the value of 100. Relative font size changes are computed relative to this base. Explicit changes result in absolute values: The headline elements `h1` to `h6` are mapped onto the values 160, 150 etc. The elements `big` and `small` increase or decrease the current size by 10. `strong` and `b` increase the font size by 5 as well, but in the context of a headline element, only by 2. Likewise, `em` and `i` cause an increase of 5. Relative and absolute font size changes realized by means of the `font` element are processed in the same manner (the analysis of CSS information is currently not supported). The image analysis processes files embedded via the `img` element. First, the values of the `src`, `height` and `width` attributes are extracted. Afterwards, the image file is transferred to the analysis machine via HTTP and its dimensions are determined and classified into several categories (explicit `height` and `width` attributes take priority over the physical dimensions): If x and y are less than 6 each, the image is used as a spacer. If x and y are between 6 and 45, we assume the image is an icon. If the quotient $\frac{x}{y} > 10$, the image acts as a separator. If the dimensions of an image are found in a list of quasi-standardised dimensions of banner advertisements (468/60, 156/60, 137/60 etc.), the image is categorised as a banner. The respective `img` element is embedded within one of the following `hypnotic:` elements: `Separator`, `Icon`, `Banner`, `Spacer`.

Stage 2: Text Encapsulation

This stage recursively processes the DOM tree and encapsulates each text node containing one or more words within the element `hypnotic:Text`. Every `Text` node receives several

attributes, especially its font size.

Stage 3: Detection of Text Blocks

The purpose of this stage is to find and to mark text blocks among the `hypnotic:Text` children found in stage 2. A text block is defined as a paragraph-like object in a more or less consistent font size. The function `findTextBlocks` utilizes a top-down/depth-first tree walker that triggers `markTextBlocks` if a text block has been found. Listing 1 shows this function in abbreviated pseudo-Perl code. Several conditions can trigger `markTextBlocks` to be called. These conditions act as boundaries between text blocks: (a) Two or more subsequent `br` elements, (b) a `hypnotic:Separator` element, (c) if the current element belongs to the elements that cause a paragraph-break in the browser (see `@b` in listing 1), (d) if the tree walker leaves an as yet unmarked subtree governed by one of the elements in `@b`, (e) a significant change in font size.

The function `markTextBlocks` takes the array consisting of one or more `hypnotic:Text` elements, under certain conditions interspersed with zero or more `hypnotic:Icon` elements and encapsulates this sequence within `hypnotic:TextBlock`. The insertion of this new element occurs as high as possible in the DOM tree, i. e., we examine the respective parent nodes of the `hypnotic:Text` elements (exploiting the path information added in stage 1) and determine whether there are additional `hypnotic:Text` children which are not part of our current sequence. If this is not the case, we examine the parent of the parent, and so on, until we find the LCN, the least common node to which we can safely attach the `hypnotic:TextBlock` node. Under certain conditions, there is no LCN. In that case, we have to include the new node at the correct position at the child level of one of the common ancestors of the elements in `@s`. We are then able to move all the subtrees that include the elements in `@s` beneath the newly inserted node. Fig. 1 shows a DOM tree modified in this manner.

Stage 4: Processing of Text Blocks

Stage 3 inserts a new level of explicit structure into the document which is only implicitly contained in the source document. Stage 4 further exploits the newly introduced structure by examining the individual subtrees governed by the `hypnotic:TextBlock` nodes in order to detect an even higher level of structure, aggregating one or more `TextBlock` nodes. At the moment, our prototype is able to detect headlines on various levels, 'footnotes' (i. e., one or more subsequent text blocks with a significantly smaller font size than the base font size) and several different types of explicit and implicit list structures.

Detecting explicitly marked lists is straightforward: Using XPath expressions, we locate all the `hypnotic:TextBlock` nodes that govern a `ul`, `ol`, or `dl` node which contains at least one `li` or `dt`/`dd` element, and encapsulate each of the resulting nodes with a separate `hypnotic:List` element. A special case exists for those lists that contain multiple text blocks (e. g., several `li` elements with explicit paragraph boundaries such as `<br><br>`). Detecting implicit lists is a rather complex process carried out by means of another top-

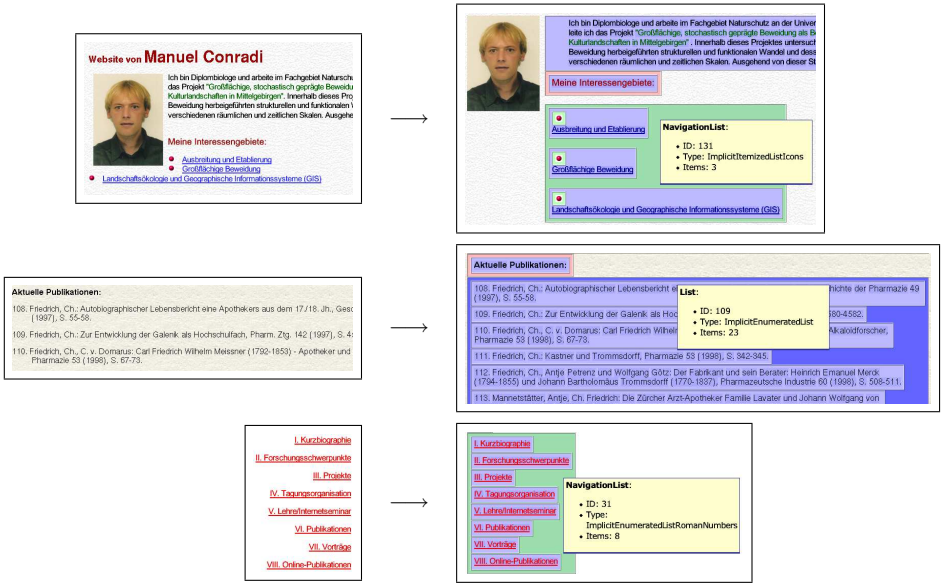


Figure 2: Examples for the detection of different types of implicit lists (see the popup-windows on the right; left: original document as rendered in Mozilla, right: visualization of the analysis in the front end of our corpus-database, see section 4.3).

down/depth-first tree walker which encapsulates certain `hypnotic:TextBlock` sequences as `hypnotic:List`, in some cases interspersed with `hypnotic:Icon`.

Currently, we detect (a) itemized lists in which the bullet point is realized by a small-sized inline image (i. e., sequences of the following structure: `Icon`, 1..*n* `TextBlock`, `Icon`, 1..*n* `TextBlock`, ...), (b) implicit itemized lists in which the bullet point is simulated by typographical means (e. g., a consistent - or * at the beginning of each `TextBlock`), (c) implicit enumerated lists in which the label consists of numbers of increasing values in arabic or (d) roman notation (see fig. 2 for examples). Again, the insertion of the new `List` node uses the algorithm to find the LCN of a given sequence of nodes in order to locate the correct point within the DOM tree (see stage 3). Up to this point, we do not take into account any hyperlinks contained within a `List`.

The headline and footnote detection operates on the font sizes of adjacent text blocks. If these are significantly higher or lower than the base font size, the respective sequence is marked as `Headline` or `Footnote`. Both elements receive an attribute `Level`, marking the relative headline or footnote level (based on the font size). In addition, we detect headlines embedded in text blocks: As the font size within a text block may vary with

a threshold of ± 5, an initial `Text` node with a font size of +5 is marked as an inline `Headline`.

Stage 5: Detecting Different Types of Hyperlink Lists

In the fifth stage several heuristics detect different types of hyperlink lists. For this purpose, the parser determines all instances of `hypnotic:List` that have at least one list item containing a hyperlink. As the information about the number of hyperlinks comprised in a list as well as the different types of hyperlinks are already available, the identification process can be based on the percentage of a hyperlink type: if more than two thirds of all hyperlinks belong to the type `internal`, we assume that these hyperlinks connect nodes of the current hypertext and replace the element `hypnotic:List` by `hypnotic:NavigationList`. Similar replacement rules exist for the remaining types `external` (creates `hypnotic:Hotlist`), `thispage` (`hypnotic:TableOfContents`) and `samedomain` (`hypnotic:Dispenser`, a special type of navigation list; see figure 2 for examples).

Stage 6: Integrating Part-of-Speech-Information

The final stage incorporates a robust syntactic parser. By means of an XPath expression, all `hypnotic:TextBlock` elements are determined. In all of these elements, all `hypnotic:Text` elements are collected. These sequences of characters are used to construct an array whose fields contain whitespace-separated tokens. This array is passed on to the commercial POS tagger and syntactic parser Connexor Machinese Syntax German that returns the analysis result in the form of an XML document. As this tool's tokenisation algorithm is not documented, it is necessary to map the Connexor tokenisation (e. g. "[Paul] [laughs] [.]") to our whitespace-separated token approach ("[Paul] [laughs.]"), because a new `hypnotic:Token` element containing the POS information as an attribute value is constructed for *every* word in the DOM structure.

4.3 Visualization and Examples

The corpus-database and the text parser are accessible by means of a web front end implemented in PHP, which accesses the MySQL database in order to retrieve and display the locally stored documents in a browser (see fig. 4). After activation, the source document is processed by the text parser, and the analysis result, i. e., the augmented and serialized DOM tree, is sent through an XSLT stylesheet which transforms the elements of the `hypnotic:` namespace into tables with specific background colours, so that the results can be conveniently displayed within the front end itself (fig. 2 shows three excerpts of these before/after examples, fig. 4 shows a complete example). Furthermore, the XSLT stylesheet converts the analysis information contained within the attributes of the `Headline`, `List` etc. elements so that they are displayed in a popup-window as soon as the mouse pointer is moved over the respective table region (again, see fig. 2); the required JavaScript code fragments are generated dynamically. For the development

phase of the text parser, this function is a clear advantage, as it is far more user-friendly than examining large amounts of raw, tagged XML markup, e. g., in an XML editor. Furthermore, in some situations, the text parser tends recursively to nest certain elements within each other, e. g., a `TextBlock` within a `TextBlock`. Annotation errors like these can be detected extremely quickly by means of corresponding templates in the XSLT stylesheet: if such an automatic error detection template matches (in this case, `hypnotic:TextBlock//hypnotic:TextBlock`), the stylesheet prints an error message at the beginning of the result document, indicating that something went wrong.

In addition, the web front end provides functions for rendering the source document, the augmented document (see fig. 1), and an HTML-free version of the augmented structure as DOM-like trees (in Postscript or GIF format). These functions are important for verifying that the insertion of `hypnotic:*` nodes works correctly. The reduced version of the augmented structure is realized by means of another XSLT stylesheet that removes all elements of the `xhtml:` namespace (see fig. 3). In order not to obtain a completely flat tree, intermediate nodes (e. g., a `xhtml:p` node that governs two text block nodes) are substituted by a dummy node (`hypnotic:Node`, see the tree on the right hand side in fig. 3).

4.4 Implementation

The text parser is implemented in Perl and uses the modules XML::LibXML (i. e., the Gnome project's libxml2 library providing a DOM Level 2 parser and an XPath engine), XML::LibXSLT (i. e., Gnome's libxslt library). Furthermore, LWP::Simple is used to establish HTTP connections, and Image::Size is utilised to extract the physical dimensions of image files. The automatic generation of DOM trees is based on the GraphViz tree drawing package and a customized version of the Perl module GraphViz::XML.

4.5 Towards Automatic Web Genre Identification

Section 3 describes our theoretical framework and briefly mentions our ultimate goal: devising a web genre-enabled search engine. Our current design for the architecture of this search engine is built on three main components (Rehm, 2005): (a) The text parser is needed to extract as much explicitly- and implicitly-contained structure in a source document as possible. This data, along with certain keywords will comprise the majority of information which is required for, (b) the web genre classifier, which, in our opinion, can be based on one (or more) of the classic machine learning algorithms or approaches (*k*NN, Naive Bayes, C 4.5 etc.). The third component is (c) a set of ontologies/taxonomies, represented in OWL, that specify information to be used in (i) the classification task (e. g., a web genre ontology) and, (ii) the next level of automatic annotation: As soon as the system is able to automatically classify an arbitrary web page of our domain of interest into its respective web genre, we intend to tackle the problem of automatically converting the document into a markup language that explicates the content and structure of the respective web genre. In other words, we want to map individual text blocks (and higher

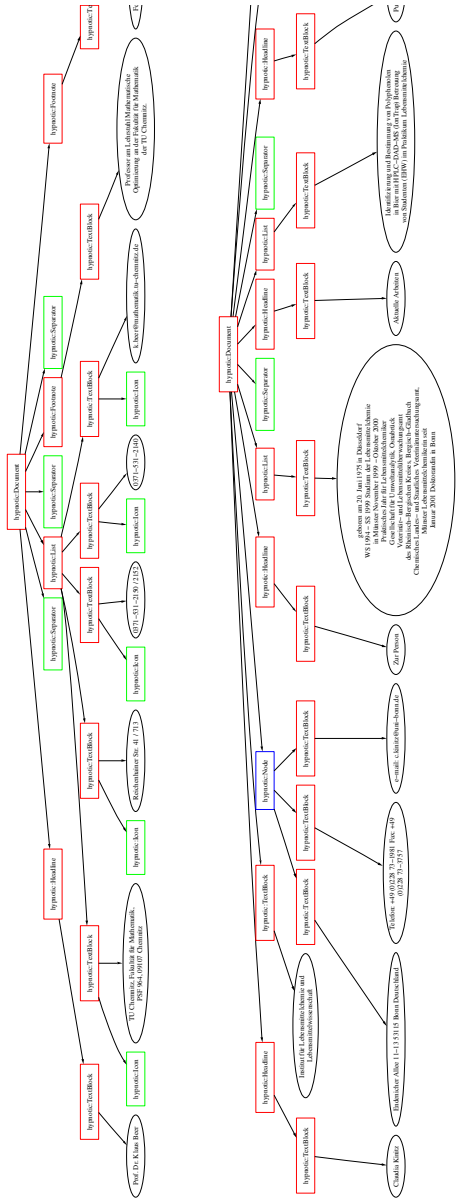


Figure 3: Left: The reduced DOM structure of the document shown in fig. 4. Right: This reduced tree shows that we capture the original HTML-based hierarchical structure by inserting a dummy node (`hypnotic:Node`).

Figure 4: The web front end of the corpus-database in “document view” mode (top: source document, bottom: document as analysed by the text parser and visualized by an XSLT stylesheet; colour key: light blue – text block, dark blue – list, red – separator, light red – headline, green – footnote, light green – icon).

level objects) onto web genre modules. A function like this would, in turn, enable a novel kind of information extraction application; the extraction of the contents of specific web genre modules of web pages that belong to a certain web genre. For these purposes, we envision a comprehensive OWL-/RDF-based format such as the one presented in Potok et al. (2002) to encapsulate the information needed for the mapping algorithm.

5 Conclusions and Future Work

The algorithms of the text parser are quite stable and robust. Of the 100 documents contained in our document sample testbed, only four documents are incorrectly annotated. These documents contain very unusual markup structures which, as yet, cannot be rearranged by the `markTextBlock` algorithm. We will enhance the algorithms as necessary in the near future, so that the complete sample can be analysed correctly. Most documents contained in the sample use `table` elements to realise highly complex layouts. In certain cases, our algorithms are not able correctly to detect the structure contained in these tables. We consciously ignored this problem as we plan to enhance the cascaded processing stages by one additional stage which has to run after the completion of stage 2, in order to (a) distinguish genuine tables from layout-oriented tables, and, to (b) classify each layout-oriented table into the correct form (see Hurst, 2002, Wang et al., 2000, and the other approaches cited in section 2). Furthermore, additional extensions will be implemented to detect other implicit structures and higher level objects. We will add heuristics to examine the contents of text blocks in order to distinguish between text blocks containing only text fragments (such as all the text blocks shown in fig. 4) and those containing continuous text. After mapping each text block into one of these two classes, we can add another layer of processing by identifying individual sentences and annotating the analysis by augmenting the XHTML/XML markup with explicit sentence boundaries.

Apart from the web genre identification application (see section 4.5), our text parser could be used for several other tasks: the hierarchical information which is contained in the individual elements (headline, footnote, text block, list etc.) as attributes, could be used to rearrange the reduced DOM structure in a way that reflects this very hierarchical structuring. A function like this could be used to facilitate browsing on mobile devices (e. g., to display only the headlines first, so that the user can decide which headline to explore further by dynamically folding out the subtree governed by the headline node), or as an initial stage for automatic text summarization approaches, especially in conjunction with the more detailed text block analysis sketched in the previous paragraph. Furthermore, the envisioned automatic sentence boundary detection could be used to gather collections of test sentences for natural language parser research and evaluation (Web as Corpus approach, Kilgarriff, 2001, Rehm, 2003). Another possible application for the text parser is the automatic weighting of certain text fragments within a traditional search engine context: a specific keyword contained in a first-level-headline is definitely more important to a certain web page than the same keyword contained in a 3rd-level-footnote.

References

Alam, H., F. Rahman, Y. Tarnikova, and A. Kumar (2003). When is a List is a List?: Web Page Re-authoring for Small Display Devices. In *Proceedings of WWW 2003*, Budapest.

Barnard, D. T., L. Burnard, S. J. DeRose, D. G. Durand, and C. Sperberg-McQueen (1996). Lessons for the World Wide Web from the Text Encoding Initiative. *The World Wide Web Journal 1*(1), 349–357.

Buyukkokten, O., H. Garcia-Molina, and A. Paepcke (2001). Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In *Proceedings of WWW 2001*, Hong Kong.

Carchiolo, V., A. Longheu, and M. Malgeri (2003). Extracting Logical Schema from the Web. *Applied Intelligence 18*, 341–355.

Chan, M. and G. Yu (1999). Extracting Web Design Knowledge: The Web De-Compiler. In *IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS 1999)*, Volume 2, Florence, pp. 547–552.

Chen, J., B. Zhou, J. Shi, H. Zhang, and Q. Fengwu (2001). Function-Based Object Model Towards Website Adaption. In *Proceedings of WWW-10*, Hong Kong, pp. 587–596.

Chen, Y., W.-Y. Ma, and H.-J. Zhang (2003). Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *Proceedings of WWW 2003*, Budapest.

Chung, C. Y., M. Gertz, and N. Sunaresan (2002). Reverse Engineering for Web Data: From Visual to Semantic Structures. In *Proceedings of the 18th Int. Workshop on Data Engineering (ICDE '02)*, San Jose, pp. 53–63.

Chung, C. Y., M. Gertz, and N. Sundaresan (2001). Quixote: Building XML Repositories from Topic Specific Web Documents. In J. S. Giansalvatore Mecca (Ed.), *Fourth Int. Workshop on the Web and Databases (WebDB 2001)*, Santa Barbara, pp. 103–108.

Clark, J. (1999). XSL Transformations (Version 1.0). Technical Specification, W3C. `http://www.w3.org/TR/xslt/`.

Clark, J. and S. DeRose (1999). XML Path Language (XPath) – Version 1.0. Technical Specification, W3C. `http://www.w3.org/TR/xslt/`.

Cohen, W. W., M. Hurst, and L. S. Jensen (2002). A Flexible Learning System for Wrapping Tables and Lists in HTML Documents. In *Proceedings of WWW 2002*, Honolulu.

Cowie, J. and Y. Wilks (2000). Information Extraction. In R. Dale, H. Moisl, and H. Somers (Eds.), *Handbook of Natural Language Processing*, pp. 241–260. New York, Basel: Marcel Dekker.

Crowston, K. and M. Williams (1997). Reproduced and Emergent Genres of Communication on the World-Wide Web. In *Proc. of the 30th Hawaii Int. Conf. on Systems Sciences (HICSS-30)*, Volume 6, pp. 30–39.

de Saint-Georges, I. (1998). Click Here if You Want to Know Who I Am. Deixis in Personal Homepages. In *Proceedings of the 31st Hawaii Int. Conf. on Systems Sciences (HICSS-31)*, Volume 2, pp. 68–77.

Dillon, A. and B. A. Gushrowski (2000). Genres and the Web: Is the Personal Home Page the First Uniquely Digital Genre? *Journal of the American Society for Information Science 51*(2), 202–205.

Eiron, N. and K. S. McCurley (2003). Untangling Compound Documents on the Web. In *Proceedings of the 14th ACM Conf. on Hypertext and Hypermedia*, pp. 85–94. Nottingham.

Fielding, R. T., J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, and P. J. L. T. Berners-Lee (1999). Hypertext Transfer Protocol – HTTP/1.1. Network Working Group – Request for Comments (RFC). Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach und Tim Berners-Lee. `http://www.ietf.org/rfc/`.

Furuta, R. and C. C. Marshall (1996). Genre as Reflection of Technology in the World-Wide Web. In S. Fraïssé, F. Garzotto, T. Isakowitz, J. Nanard, and M. Nanard (Eds.), *Hypermedia Design, Proc. of the Int. Workshop on Hypermedia Design (IWHD 1995)*, pp. 182–195. Berlin, Heidelberg, New York etc.: Springer.

Gupta, S., G. Kaiser, D. Neistadt, and P. Grimm (2003). DOM-based Content Extraction of HTML Documents. In *Proceedings of WWW 2003*, Budapest.

Haas, S. W. and E. S. Grams (2000). Readers, Authors, and Page Structure – A Discussion of Four Questions Arising from a Content Analysis of Web Pages. *Journal of the American Society for Information Science 51*(2), 181–192.

Haase, M., M. Huber, A. Krumeich, and G. Rehm (1997). Internetkommunikation und Sprachwandel. In R. Weingarten (Ed.), *Sprachwandel durch Computer*, pp. 51–85. Opladen: Westdeutscher Verlag.

Hors, A. L., P. L. Hégaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne (2000). Document Object Model (DOM) Level 2 Core Specification. Technical Specification, W3C.

Hurst, M. (2002). Classifying TABLE Elements in HTML. In *Proceedings of WWW 2002*, Honolulu.

Kilgarriff, A. (2001). Web as Corpus. In P. Rayson, A. Wilson, T. McEnery, A. Hardie, and S. Khoja (Eds.), *Proceedings of the Corpus Linguistics 2001 Conf.*, Lancaster, pp. 342–344.

Lim, S.-J. and Y.-K. Ng (1999). An Automated Approach for Retrieving Hierarchical Data from HTML Tables. In *Proceedings of the 8th Int. Conf. on Information and Knowledge Management (CIKM '99)*, pp. 466–474. ACM Press.

Mehler, A., M. Dehmer, and R. Gleim (2004). Towards Logical Hypertext Structure – A Graph-Theoretic Perspective. In T. Böhme and G. Heyer (Eds.), *Proceedings of the Fourth Int. Workshop on Innovative Internet Computing Systems (I2CS '04)*, Lecture Notes in Computer Science, Berlin, New York. Springer.

Myllymaki, J. (2001). Effective Web Data Extraction with Standard XML Technologies. In *Proceedings of WWW-10*, Hong Kong, pp. 689–696.

Penn, G., J. Hu, H. Luo, and R. McDonald (2001). Flexible Web Document Analysis for Delivery to Narrow-Bandwidth Devices. In *Int. Conf. on Document Analysis and Recognition (ICDAR '01)*, Seattle, pp. 1074–1078.

Potok, T. E., M. T. Elmore, J. W. Reed, and N. F. Samatova (2002). An Ontology-based HTML to XML Conversion Using Intelligent Agents. In *Proceedings of the 35th Hawaii Int. Conf. on System Sciences (HICSS-35)*, Big Island, Hawaii.

Raggett, D., A. L. Hors, and I. Jacbos (1999). HTML 4.01 Specification. Technical Specification, W3C. `http://www.w3.org/TR/html401/`.

Rehm, G. (2001). `korpus.html` – Zur Sammlung, Datenbank-basierten Erfassung, Annotation und Auswertung von HTML-Dokumenten. In H. Lobin (Ed.), *Proceedings of the GLDV Spring Meeting 2001*, Giessen, Germany, pp. 93–103. Gesellschaft für linguistische Datenverarbeitung. `http://www.uni-giessen.de/fbo9/ascl/gldv2001/`.

Rehm, G. (2002). Towards Automatic Web Genre Identification – A Corpus-Based Approach in the Domain of Academia by Example of the Academic's Personal Homepage. In *Proceedings of the 35th Hawaii Int. Conf. on System Sciences (HICSS-35)*, Big Island, Hawaii.

Rehm, G. (2003). Texttechnologie und das World Wide Web – Anwendungen und Perspektiven. In H. Lobin and L. Lemnitzer (Eds.), *Texttechnologie – Anwendungen und Perspektiven*, Stauffenburg Handbücher, pp. 433–464. Tübingen: Stauffenburg.

Rehm, G. (2004a). Hypertextsorten-Klassifikation als Grundlage generischer Informationsextraktion. In A. Mehler and H. Lobin (Eds.), *Automatische Textanalyse – Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, pp. 219–233. Wiesbaden: Verlag für Sozialwissenschaften.

Rehm, G. (2004b). Ontologie-basierte Hypertextsorten-Klassifikation. In A. Mehler and H. Lobin (Eds.), *Automatische Textanalyse – Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, pp. 121–137. Wiesbaden: Verlag für Sozialwissenschaften.

Rehm, G. (2005). *Hypertextsorten: Definition – Struktur – Klassifikation*. Ph. D. thesis, Institut für Germanistik, Angewandte Sprachwissenschaft und Computerlinguistik, Justus-Liebig-Universität Gieß en.

Shepherd, M. and C. Watters (1999). The Functionality Attribute of Cybergenres. In *Proceedings of the 32nd Hawaii Int. Conf. on Systems Sciences (HICSS-32)*.

Song, R., H. Liu, J.-R. Wen, and W.-Y. Ma (2004). Learning Block Importance Models for Web Pages. In *Proceedings of WWW-2004*, New York, pp. 203–211. ACM Press. Refereed Papers Track.

Wang, H.-L., S.-H. Wu, I. C. Wang, C.-L. Sung, W. L. Hsu, and W. K. Shih (2000). Semantic Search on Internet Tabular Information Extraction for Answering Queries. In *Proceedings of the 9th Int. Conf. on Information and Knowledge Management (CIKM 2000)*, McLean, pp. 243–249. ACM Press.

Wang, Y. and J. Hu (2002). A Machine Learning Based Approach for Table Detection on The Web. In *Proceedings of WWW 2002*, Honolulu.

Yang, G., S. Mukherjee, W. Tan, I. V. Ramakrishnan, and H. Davulcu (2003). On the Power of Semantic Partitioning of Web Documents. In S. Kambhampati and C. A. Knoblock (Eds.), *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, Acapulco, pp. 21–26.

Chris Biemann

Ontology Learning from Text: A Survey of Methods

1 Introduction

After the vision of the Semantic Web was broadcasted at the turn of the millennium, *ontology* became a synonym for the solution to many problems concerning the fact that computers do not understand human language: if there were an ontology and every document were marked up with it and we had agents that would understand the mark-up, then computers would finally be able to process our queries in a really sophisticated way. Some years later, the success of Google shows us that the vision has not come true, being hampered by the incredible amount of extra work required for the intellectual encoding of semantic mark-up – as compared to simply uploading an HTML page. To alleviate this acquisition bottleneck, the field of ontology learning has since emerged as an important sub-field of ontology engineering.

It is widely accepted that ontologies can facilitate text understanding and automatic processing of textual resources. Moving from words to concepts not only mitigates data sparseness issues, but also promises appealing solutions to polysemy and homonymy by finding non-ambiguous concepts that may map to various realizations in – possibly ambiguous – words.

Numerous applications using lexical-semantic databases like WordNet (Miller, 1990) and its non-English counterparts, e.g. EuroWordNet (Vossen, 1997) or CoreNet (Choi and Bae, 2004) demonstrate the utility of semantic resources for natural language processing.

Learning semantic resources from text instead of manually creating them might be dangerous in terms of correctness, but has undeniable advantages: Creating resources for text processing from the texts to be processed will fit the semantic component neatly and directly to them, which will never be possible with general-purpose resources. Further, the cost per entry is greatly reduced, giving rise to much larger resources than an advocate of a manual approach could ever afford. On the other hand, none of the methods used today are good enough for creating semantic resources of any kind in a completely unsupervised fashion, albeit automatic methods can facilitate manual construction to a large extent.

The term *ontology* is understood in a variety of ways and has been used in philosophy for many centuries. In contrast, the notion of ontology in the field of computer science is younger – but almost used as inconsistently, when it comes to the details of the definition.

The intention of this essay is to give an overview of different methods that learn ontologies or ontology-like structures from unstructured text. Ontology learning from other sources, issues in description languages, ontology editors, ontology merging and ontology evolving transcend the scope of this article. Surveys on ontology learning from text and other sources can be found in Ding and Foo (2002) and Gómez-Pérez

and Manzano-Macho (2003), for a survey of ontology learning from the Semantic Web perspective the reader is referred to Omelayenko (2001).

Another goal of this essay is to clarify the notion of the term *ontology* not by defining it once and for all, but to illustrate the correspondences and differences of its usage.

In the remainder of this section, the usage of *ontology* is illustrated very briefly in the field of philosophy as contrasted to computer science, where different types of ontologies can be identified.

In section 2, a variety of methods for learning ontologies from unstructured text sources are classified and explained on a conceptual level. Section 3 deals with the evaluation of automatically generated ontologies and section 4 concludes.

1.1 Ontology in philosophy

In philosophy, the term *ontology* refers to the study of existence. In this sense, the subject is already a central topic of *Aristotle's Categories* and in all metaphysics. The term was introduced in the later Renaissance period, see Ritter and Gründer (1995), as "*lat.* philosophia de ente". In the course of centuries, *ontology* was specified in different ways and covered various aspects of metaphysics. It was sometimes even used as a synonym for this field. Further, the distinction between ontology and theology was not at all times clear and began to emerge in the 17th century.

For Leibniz, the subject of *ontology* is everything that can be recognized (*germ.* erkannt). Recognition (*germ.* Erkenntnis) as a basis of metaphysics is criticised by Kant, who restricts ontology to a propaedeutical element of metaphysics, containing the conditions and the most fundamental elements of all our recognition (*germ.* Erkenntniß) a priori.

The relation of ontology to logic was introduced by Hegel and later strengthened by Husserl, who defends the objectivity of logical entities against subjectivation and replaces the notion of logical terms as psychical constructions with "ideal units" that exist a priori. Ontology in this context can be divided into two kinds: *formal ontology* that constitutes itself as a theory of all possible forms of theories, serving as science of sciences, and *regional* or *material ontologies* that are the a priori foundations of empirical sciences (Husserl, 1975). The latter notion paved the way to *domain-specific ontologies*, see section 1.2.

For computer science, the most influential definition has been given by Quine (cf. Quine, 1969), who binds scientific theories to ontologies. As long as a theory holds (because it is fruitful), theoreticians perform an ontological commitment by accepting the a priori existence of objects necessary to prove it. A consequence of his famous quote "to be is to be the value of a bound variable" is: As long as scope and domain of quantified variables (objects) are not defined explicitly by an ontology, the meaning of a theory is fuzzy. Ontologies in the sense of Quine are the outcome of empirical theories, and hence they also need to be justified empirically.

To subsume, ontology abstracts from the observable objects in the world and deals with underlying principles of existence as such.

1.2 Ontologies in Computer Science

Ontology in computer science is understood not as general as in philosophy, because the perception of ontologies is influenced by application-based thinking. But still ontologies in computer science aim at explaining the world(s), however, instead of embracing the whole picture, they only focus on what is called a *domain*. A domain is, so to speak, the world as perceived by an application. Example: The application of a fridge is to keep its interior cold and that is reached by a cooling mechanism which is triggered by a thermostat. So the domain of the fridge consists only of the mechanism and the thermostat, not of the food in the fridge, and can be expressed formally in a fridge ontology. Whenever the application of the fridge is extended, e.g. to illuminate the interior when the door is opened, the fridge ontology has to be changed to meet the new requirements. So much about the fridge world. In real applications, domains are much more complicated and cannot be overseen at a glance.

Ontologies in computer science are specifications of shared conceptualizations of a domain of interest that are shared by a group of people. Mostly, they build upon a hierarchical backbone and can be separated into two levels: upper ontologies and domain ontologies.

Upper ontologies (or foundation ontologies), which describe the most general entities, contain very generic specifications and serve as a foundation for specializations. Two well-known upper ontologies are SUMO (Pease and Niles, 2002) and CyC (Lenat, 1995). Typical entries in upper ontologies are e.g. "entity", "object" and "situation", which subsume a large number of more specific concepts. Learning these upper levels of ontologies from text seems a very tedious, if not impossible task: The connections as expressed by upper ontologies consist of general world knowledge that is rather not acquired by language and is not explicitly lexicalized in texts.

Domain ontologies, on the other hand, aim at describing a subject domain. Entities and relations of a specific domain are sometimes expressed directly in the texts belonging to it and can eventually be extracted. In this case, two facts are advantageous for learning the ontological structures from text: The more specialized the domain, the less is the influence of word sense ambiguity according to the "one sense per domain"-assumption in analogy to the "one sense per discourse"-assumption (Gale et al., 1993). Additionally, the less common-knowledge a fact is, the more likely it is to be mentioned in textual form.

In the following section, distinctions between different kinds of ontologies and other ways of categorizing the world are drawn.

1.3 Types of Ontologies

John Sowa (Sowa, 2003) classifies ontologies into three kinds. A *formal ontology* is a conceptualization whose categories are distinguished by axioms and definitions. They are stated in logic that can support complex inferences and computations. The knowledge representation community defines ontology in accordance as follows:

> "[An ontology is] a formal, explicit specification of a shared conceptualization. 'Conceptualization' refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. 'Formal' refers to the fact that the ontology should be machine readable. 'Shared' reflects that ontology should capture consensual knowledge accepted by the communities." (Gruber, 1993; Ding and Foo, 2002)

As opposed to this, categories in *prototype-based ontologies* are distinguished by typical instances or prototypes rather than by axioms and definitions in logic. Categories are formed by collecting instances extensionally rather than describing the set of all possible instances in an intensional way, and selecting the most typical members for description. For their selection, a similarity metric on instance terms has to be defined.

The third kind of ontology are *terminological ontologies* that are partially specified by subtype-supertype relations and describe concepts by concept labels or synonyms rather than prototypical instances, but lack an axiomatic grounding. A well known example for a terminological ontology is WordNet (Miller, 1990).

Figure (1) illustrates different ontology paradigms for a toy example food domain divided into vegetarian and non-vegetarian meals.

All of these paradigms have their strengths and weaknesses. Formal ontologies directly induce an inference mechanism. Thus, properties of entities can be derived when needed. A drawback is the high effort of encoding and the danger of running into inconsistencies. Further, exact interference may become intractable in large formal ontologies.

Terminological and prototype-based ontologies cannot be used in a straightforward way for inference, but are easier to construct and to maintain. A disadvantage of the prototype-based version is the absence of concept labels, which makes it impossible to answer queries like "Tell me kinds of cheese!". Due to the absent labeling during construction, they are directly induced by term clustering and therefore easier to construct but less utilizable than their terminological counterparts.

A distinction that causes confusion are the notions of taxonomy versus ontology, which are occasionally used in an interchangeable way. Taxonomies are collections of entities ordered by a classification scheme and usually arranged hierarchically. There is only one type of relation between entries, mostly the IS-A or PART-OF relation. This corresponds to the notion of terminological ontologies. For formal ontologies, the concepts together with IS-A relations form the taxonomic backbone of the ontology.

Another kind of resource which is a stepping stone towards ontologies are thesauri like Roget's Thesaurus (Roget, 1852) for English or Dornseiff (Dornseiff, 2004) for German. A thesaurus contains sets of related terms and thus resembles a prototype-based ontology. However, different relations are mixed: a thesaurus contains hierarchy relations amongst others, but they are not marked as such.

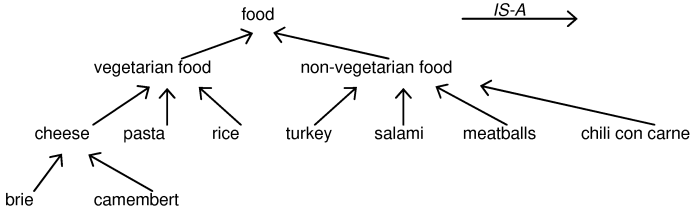
Formal ontology

Axioms:
food(brie), food(camembert), food(turkey), food(meatballs), food(chili con carne), meat(turkey), meat(minced meat), part_of(minced meat, chili con carne), part_of(minced meat, meatballs)

veg_food(x) = { x | food(x) $\wedge$ ($\neg$part_of(y,x) $\wedge$ meat(y)) $\wedge$ $\neg$meat(x) }
non_veg_food(x) = { x | food(x) $\wedge$ ((part_of(y,x) $\wedge$ meat(y)) $\vee$ meat(x)) }

Possible to derive: "turkey" and "chili con carne" are non-vegetarian foods

Terminological ontology



Prototype-based ontology

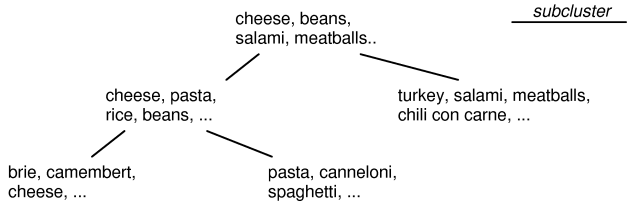


Figure 1: Formal vs. terminological vs. prototype-based food ontology.

2 Learning Ontologies from unstructured Text

Ontologies can be learnt from various sources, be it databases, structured and unstructured documents or even existing preliminaries like dictionaries, taxonomies and directories. Here, the focus is on acquisition of ontologies from unstructured text, a format that scores highest on availability but lowest on accessibility.

Most approaches use only nouns as the bricks for ontology building and disregard any ontological relations between other word classes.

To a large extent, the methods aim at constructing IS-A-related concept hierarchies rather than full-fledged formal ontologies. Other subtype-supertype relations like PART-OF are examined much less.

One underlying assumption for learning semantic properties of words from unstructured text data is Harris' distributional hypothesis (Harris, 1968), stating that similar words tend to occur in similar contexts. It gives rise to the calculation of paradigmatic relations (cf. Heyer et al., 2005), called 'associations' in de Saussure (1916). We shall see that the notion of context as well as the similarity metrics differs considerably amongst the approaches presented here.

Another important clue is the use of patterns that explicitly grasp a certain relation between words. After the author who introduced patterns such as "*X*, *Y*s and other *Z*s" or "*W*s such as *X*, *Y* and *Z*", they are often referred to as Hearst-patterns (Hearst, 1992), originally used to extract IS-A relations from an encyclopedia for the purpose of extending WordNet. Berland and Charniak (1999) use similar kinds of patterns to find instances of the PART-OF relation.

As learning from text usually involves statistics and a corpus, using the world wide web either as additional resource or as the main source of information is often a possibility to avoid data sparseness as discussed in Keller et al. (2002) and carried out e.g. by Agirre et al. (2000) and Cimiano and Staab (2004).

Ontology learning techniques can be divided in constructing ontologies from scratch and extending existent ontologies. The former comprises mostly clustering methods that will be described in section 2.1, the latter is a classification task and will be treated in section 2.2. Approximately, this is the distinction between unsupervised versus supervised methods, although we shall see that some clustering approaches involve supervision in intermediate steps.

Section 2.3 summarizes research undertaken in semantic lexicon construction, which is a related task to ontology learning, although the representation of results might differ. In section 2.4, the view of ontology learning as an Information Extraction exercise is discussed.

2.1 Clustering for Ontology Learning

In hierarchical clustering, sets of terms are organized in a hierarchy that can be transformed directly into a prototype-based ontology. For clustering, a distance measure on terms has to be defined that serves as the criterion for merging terms or clusters of terms. The same measure can be used – if desired – to compute the most typical instances of a concept as the ones closest to the centroid (the hypothetical 'average' instance of a set). Crucial to the success of this methodology is the selection of an appropriate measure of semantic distance and a suitable clustering algorithm.

An overview of clustering methods for obtaining ontologies from different sources including free text can be found in Maedche and Staab (2004). In principle, all kinds of clustering methods – be it agglomerative or divisive – can be applied to all kinds of representations, be it vector space (Salton et al., 1975), associative networks (Heyer and Witschel, 2005) or set-theoretic approaches as presented in Cimiano et al. (2004). Here, the focus will be on just a few, illustrative methods.

Methods based on distributional similarity Methods using distributional similarity can be divided into syntactic and window-based approaches.

Syntactic approaches make use of similarity regarding predicate-argument relations (i.e. verb-subject and verb-object relations), the usage of adjective modifiers or subjective predicates is rare.

An early paper on semantic clustering is Hindle (1990), which aims at finding semantically similar nouns by comparing their behavior with respect to predicate-argument structures. For each verb-subject and verb-object pair in his parsed 6 million word corpus, he calculates co-occurrence weights as the mutual information within the pairs. Verb-wise similarity of two nouns is the minimum shared weight, and the similarity of two nouns is the sum of all verb-wise similarities. The exemplified analysis of this similarity measure exhibits mostly homogeneous clusters of nouns that act or are used in a common way.

For obtaining noun hierarchies from text, Pereira et al. (1993) chose an encyclopedia as a well-suited textual resource for a divisive clustering approach based on verb-object relations, allowing the nouns to be members in multiple clusters.

A whole class of syntactic approaches is subsumed in the Mo'K workbench (Bisson et al., 2000), which provides a framework to define hierarchical term clustering methods based on similarity in contexts limited to specific syntactic constructions. In the same work, comparative studies between different variants of this class are presented, including ASIUM (Faure and Nédellec, 1998; Dagan et al., 1994). Another paper on using selectional preferences is e.g. Wagner (2000).

A different direction is using methods that produce paradigmatic relations as candidate extraction mechanism without syntactic pre-processing. A well-known source of paradigmatic relations is the calculation of second-order co-occurrences, which does not rely on parsing. While (first-order) co-occurrences rate pairs of word high that occur together often in a certain text window, second order co-occurrences are words that have similar distributions of first-order co-occurrences (see e.g. Ruge (1992), Schütze (1998), Rapp (2002), Biemann et al. (2004) – this corresponds roughly to Rieger's δ-abstraction (Rieger, 1981; Leopold, 2005)). The context definition of these methods is mostly not restricted to any syntactic construction, which introduces more noise but keeps the method language-independent. It can be argued that given a sufficient corpus size, equal results to syntactically aided methods might be achieved, see e.g. Pantel et al. (2004). However, as the underlying bag-of-words simplification of window-based methods abstracts from the order of the words, no clues for the relation between candidate pairs can be drawn directly from this data, making these approaches on their own not viable for the construction of ontologies from scratch.

While there does not seem to be an alternative to use patterns in order to alleviate the labeling problem, the action of naming super-concepts is not necessary when aiming at a prototypical ontology, such as in Paaß et al. (2004): here, a hierarchical extension to Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) is introduced. PLSA (like LSA – (cf. Deerwester et al., 1990)) assumes latent concepts, which are playing the role of an intermediate concept layer: the probability of seeing a word w in a document d is

the sum of the product probabilities of d belonging to concepts c and w being generated when c is present. To introduce hierarchical dependencies, the probability mass is split between sub- and super-concepts. In an experiment, a fixed 4-level hierarchy with 1, 14, 28 and 56 nodes on the levels was defined. The words with the highest probability per concept constitute the entries of the prototypical ontology. While results look impressive, a clear drawback is the predefined structure of the hierarchy.

Methods based on extraction patterns The other possibility is to use explicit clues, like Hearst-patterns.

Caraballo (1999) constructs a terminological ontology from text in the following way: noun candidates from a newspaper corpus are obtained by considering conjunction and appositive data. For all nouns, a co-occurrence matrix is set up. Similarity between two nouns is calculated by computing the cosine between their respective vectors and used for hierarchical bottom-up clustering. For labelling this hierarchy in a post-processing step, Hearst-patterns are used for finding hypernym candidates, which are placed as common parent nodes for clusters, if appropriate. Evaluated by human judgement, the method performs at about 35-55% precision.

A similar approach is presented by Cimiano and Staab (2005) who also cluster nouns based on distributional similarity and use Hearst-patterns, WordNet and patterns on the web as a hypernym oracle for constructing a hierarchy. Unlike as in Caraballo (1999), the hypernym sources are directly integrated into the clustering, deciding for each pair of nouns how they should be arranged into the hierarchy. The resulting taxonomy outperforms Caraballo's when evaluating the outcome against a reference ontology (see section 3).

2.2 OL as a classification task

Given an existing ontology, its extension can be viewed as a classification task: features of the existing data are used as a training set for Machine Learning, which produces a classifier for previously unknown instances.

One possibility is to utilize the hierarchical structure in a decision tree, as proposed in Alfonseca and Manandhar (2002). When inserting new concepts, it is tested whether they fit best to the actual node or one of the daughter nodes. The tree is traversed top-down from the root until an appropriate position is found. The largest problem here is the general nature of top-level concepts that leads to taking the wrong path in the beginning of the process, which can be alleviated by propagating the signatures of lower-level concepts one step upwards. For around 1200 concepts, an accuracy of about 28% is reported. A related approach is Witschel (2005), which substitutes the syntactic dependencies for similarity by comparing words only on sentence-based co-occurrences. A small sub-tree of an existing WordNet-like hierarchy is used as training and test data. Propagating the semantic descriptions iteratively upwards to the root, the approach is biased towards putting new words into larger sub-trees. While Witschel's results are better, this might be due to the smaller number of concept classes.

In Fleischman and Hovy (2002), only eight categories for named entities denoting persons are considered. They examine five machine learning approaches on features based on preceding and following word N-grams which are combined into concepts using WordNet, reporting 70% accuracy.

Placing words into WordNet where the concentration of words with similar distributional characteristics is highest is conducted by Widdows (2003). He arrives at about 80% precision for common nouns, 34% for proper nouns and 65% for verbs.

How to enlarge WordNet by assigning appropriate named entities to the leaves using the Google index is discussed in Paşca (2005).

2.3 Ontology Learning as Semantic Lexicon Construction

The similarities between the construction of semantic lexicons and lexical ontologies – be it terminological or prototype-based – are striking. Both encode semantic similarities between terms and they abstract terms to concepts. Whereas semantic lexicons often attach semantic categories to words and do not structure the set of words internally any further (although semantic lexicons like e.g. HaGenLex (Helbig, 2001) are organized in a flat hierarchy of categories), ontologies aim at explaining all possible relations between concepts, being more fine-grained. Nevertheless, words with the same semantic label should be found in the same region of the ontology, which makes part of the methodology for automatic construction applicable to both tasks.

Let us assume that we have a small semantic lexicon, given by a set of categories, which are formed each by a set of words. Using a text corpus, we want to extend this lexicon.

Mostly, bootstrapping approaches have been used to tackle this problem. On the one hand, because bootstrapping can iteratively use previously learnt examples, which reduces the minimal size of the seed lexicon. On the other hand it does not necessarily need negative examples for learning, making the procedure viable for learning single semantic categories. The largest problem that bootstrapping methods have to face is error-propagation: misclassified items will lead to the acquisition of even more misclassified items. Various attempts have been made to minimize this thread.

In general, bootstrapping starts with a small set of seeds as current category. For every candidate item, the similarity to the current category is computed and the most similar candidates are added to the current category. These steps are conducted iteratively until a stopping criterion holds; sometimes the process is deliberately stopped after about 50-200 iterations.

Riloff and Shepherd (1997) were the first to apply bootstrapping for building semantic lexicons, extending one category at a time. Their context definition is one noun to the left and one noun to the right for head nouns in sentences. Collecting the contexts of the current category set, they calculate a score for each word by checking the relative frequency of the word appearing in the category's contexts. Amongst the first 100 words retrieved by the algorithm for categories of a seed size around 50, about 25% were judged correct by human decision. In Riloff and Jones (1999) not only classes are

assigned to words, but also the confidence of contexts supporting a class is estimated. Contexts in this work are patterns such as ''headquartered in <x>'' or ''to occupy <x>''. Moreover, only the top 5 candidates are added to the knowledge base per step, alleviating error-propagation to a precision of about 46%-76% after 50 iterations. Further improvement was gained in Thelen and Riloff (2002), where multiple categories are learned at the same time to avoid too large single categories consisting of a mixture with several other categories. In that way, about 80% accuracy for the first couple of hundred new words can be reached. This complies well with the structuralist notion of semantics being defined in a negative way (de Saussure, 1916; Eco, 1977): A category "grows" in the space of meaning as long as it meets the border of another category.

Building multiple categories simultaneously is also used in Biemann and Osswald (2005), who extend a semantic lexicon for the use of semantic parsing. As contexts, only modifying adjectives of nouns are taken into account. Semantic classes of known nouns are inherited via the modifying adjectives to previously unclassified nouns. In experiments using a co-occurrence significance measure to consider merely typical modifiers, the authors report to succeed in doubling their resource of 6000 nouns in 50 categories with an accuracy of about 80%.

As opposed to these shallow approaches, Roark and Charniak (1998) look for words occurring together in syntactical formations that involve full parsing of the corpus. A radical break with syntactical pre-processing is conducted in Biemann et al. (2004), where a lexical-semantic resource is extended without using any tagging or parsing, merely by using sentence-based co-occurrence statistics. A word is added to a category if many words of the category occur with it within a sentence window. While scores are differing strongly for selected categories, the approach serves as a language-independent baseline.

2.4 Information Extraction for Ontology Population

In Information Extraction (IE, see Grishman (1997) for a survey), templates containing roles, relations, temporal and time information to describe possible situations are encoded. The task is to fill the templates' slots by extracting relevant data from documents. IE proceeds in a situative way: instantiated templates are attached to the texts from which they have been extracted. Ontologies, on the other hand, encode conceptualizations that are not bound to specific text snippets but apply in general. Nevertheless, templates can be defined in IE systems like GATE (Bontcheva et al., 2004) and the standard IE extraction mechanisms can be employed to fill these templates, producing eventually more powerful and flexible extraction rules than the patterns mentioned before.

IE systems are historically suited for the extraction of named entities. This is why they are mainly used to find instances of concepts (like chocolate companies) and relations (like employer – employee) rather than the concepts themselves: they can be better used for populating than for constructing ontologies. After collecting all the situative template instantiations, pruning has to be applied to keep only relations that occur frequently and with high confidence.

In Brin (1998), the DIPRE system is laid out that bootstraps the AUTHOR-OF relation between writers and book titles by automatically creating extraction patterns that heavily rely on HTML-tags, but also use clues from unformatted text. Using a DIPRE-like architecture, the SNOWBALL system (Agichtein and Gravano, 2000) learns patterns for free text that has been tagged by a named entity recognizer and uses them to extract instances similar to a few user-provided example tuples, never attempting to extract all the information from each document. For example, the SNOWBALL-pattern ''<LOCATION>-based <ORGANISATION>'' extracts headquarters of companies with high precision. Sufficient recall is ensured by using a large corpus.

3 Evaluation

As ontology learning just emerged recently as a field of its own, there are not many gold standards that could be used for evaluation. Further, the desired result of ontology learning is not a simple list with binary classifications, but a far more complicated structure. To make it even worse, there is "no clear set of knowledge-to-be-acquired" (Brewster et al., 2004), not even for very specialized domains. As Smith (2004) claims, there are several possibilities of conceptualizations for one domain that might differ in their usefulness for different groups of people, but not in their soundness and justification. So even if the outcome of an algorithm does not compare well with a manually built ontology, how can its quality be judged?

Of course, there is always the option of manual evaluation, with its well-known drawbacks of being subjective and time-consuming. For complicated tasks like ontology learning, a comparably low inter-annotator agreement can be expected, which in turn means that several annotators have to judge the results to arrive at consistent figures.

But maybe it is not the ontology itself that is in the focus of interest, but its application. Learning ontologies is a goal of its own, but ontologies are usually just a resource that should improve performance on NLP tasks. Measuring improvements of ontology-supported approaches depicts best the gain for the application in focus, but it unfortunately does not provide direct scores for the ontology itself.

In the remainder of this section, several possibilities to conduct an automatic evaluation on ontologies are discussed.

3.1 Evaluation against a Gold Standard

The question on how to compare two taxonomies or ontologies is first dealt with in Maedche and Staab (2002), who show ways to compare them on lexical and on conceptual level. For the lexical level, they measure the lexical overlap between the concept names in a variant-robust way. For comparing the taxonomic backbones of two ontologies, the notion of semantic cotopy is introduced. Semantic cotopy of a concept is the union of all its sub- and super-concepts, approximating its intensional semantics. The averaged taxonomical similarity is determined by the maximal overlap of semantic cotopies. Further, the authors provide ways to compare the relations of two ontologies

and evaluate their measures by an empirical study, using the tourism ontology developed within the GETESS project (Staab et al., 1999).

When aiming at taxonomy relations, it is possible to compare results of an algorithm with lexical-semantic nets like WordNet, as employed by e.g. Wagner (2000) and Witschel (2005). Yet, whenever a relation is not found in the gold standard, the algorithm might be wrong or the gold standard might be incomplete. This even holds for large resources – Roark and Charniak (1998) report that 60% of the terms generated by their semantic class learner could not be found in WordNet.

In Brewster et al. (2004) a comparison of ontologies with automatically extracted keywords from text corpora is proposed. The method measures lexical overlap as a score of how much the ontology fits the texts, but merely in a bag-of-words fashion, disregarding internal structure.

3.2 Application-based Evaluation

Recent years saw an increasing amount of research using WordNet to improve any kind of NLP application. The bulk of these applications can in turn be used for evaluating automatically created semantic resources. In the following paragraphs, setups for an application-based evaluation of ontologies are discussed.

Document clustering and classification Document similarity is usually measured by comparison of document vectors in a vector space (Salton et al., 1975), where each dimension in the space represents one term. Ambiguity and variability of natural language might cause several concepts to be mapped onto one dimension and several dimensions to be used for one concept, resulting in spurious similarity scores. This is the main motivation to use LSA (Deerwester et al., 1990), which reduces the number of dimensions by just considering main components as determined by singular value decomposition. But LSA has a number of drawbacks, including bad scalability and black-box-like latent concepts. With a domain-specific ontology, terms that are found in or around the same concept can be mapped into one dimension. On the other hand, terms that are present in many concepts due to their semantic ambiguity can be excluded or disambiguated, see next paragraph.

The clustering induced by the similarity measure can be compared to pre-categorized collections such as the Reuters corpus (Reuters Corpus, 2000). It is also possible to train a classifier and compare its performance between presence and absence of ontology information. Evaluation using document similarity will favor ontologies that keep similar terms in similar places, possibly in a flat and not very fine-grained hierarchy.

In Heinrich et al. (2005), two latent concept methods for constructing a prototype-based ontology are compared by measuring their effects on document clustering. As latent methods include the notion of a document into their models and can be applied to cluster words as well as documents, the choice seems natural. The ontology is used for dimensionality reduction in document clustering, which is compared to a gold standard.

Word sense disambiguation The task of word sense disambiguation (WSD) is to choose the appropriate sense for ambiguous words from a predefined inventory of senses. For English, WSD methods are usually evaluated on the SENSEVAL corpora (Kilgarriff, 1998), using WordNet as sense dictionary. Senses are assigned according to the ambiguous words' contexts: Either contexts are compared to glosses and terms close to the different concepts in WordNet (unsupervised WSD) or to context profiles per sense acquired from a training corpus (supervised WSD). Using WSD for the evaluation will favour ontologies that distinguish well between the different senses of words. WSD was successfully supported by semantic resources obtained from large corpora by Gliozzo et al. (2005), where terms are mapped to domains using LSA with a large number of dimensions.

Information Retrieval and Question Answering After various attempts to use query expansion methods in order to provide better coverage for information retrieval (see e.g. Ruge (1992); Stamou and Christodoulakis (2005)), this direction to improve information retrieval has been largely abandoned as it usually decreases precision too much without considerably improving recall. A possible reason is that users actually look for what they have been typing in and not for hypernyms or even synonyms. Another problem is the lack of disambiguation clues in the query which causes the query expansion mechanism to over-generate even worse.

But ontologies can be used in other parts of the retrieval process. Taxonomies that are build automatically from web data are used by Sánchez and Moreno (2005) to group query results returned by a search engine. In this case, the user's behavior of accepting or rejecting the interface is the instance of judgement. Improving question answering by overcoming the shortfalls of the bag-of-words model is the objective of e.g. Leveling and Hartrumpf (2005). Here, a semantic lexicon forms the background knowledge for semantic parsing, which yields a semantic representation much more precise than simply considering presence or absence of terms. Extending the lexicon as described in section 2.3 should result in higher performance.

Using Information Retrieval and Question Answering tasks for evaluation will promote ontologies with high coverage, as these applications are usually tested in a generic rather than in a domain-specific setting.

Co-reference Resolution The goal of co-reference resolution is to detect words that form a referent chain in a text. These chains mostly consist of pronouns, but also synonyms, hypernyms and part-whole related terms can refer to a previously mentioned entity. Co-reference resolution can be viewed as the classification task of finding the right antecedent for a referent using e.g. grammatical, contextual and morphological features. The evaluation framework for English co-reference resolution, which is not an application itself but rather a pre-processing step for methods like summarization, abstracting and information extraction, are the MUC-6 and MUC-7 corpora (Chinchor, 1998). The use of semantic resources, however, is scarcely encountered for co-reference or anaphora resolution. An exception is Hoste (2005), where WordNet and the Dutch

part of EuroWordNet are used for additional features, which bring about only a small gain because of lacking coverage. At first glance, it seems that ontologies can only support co-reference resolution in the rare cases of nouns referring to other nouns that are semantically related, but not in the default case of pronouns referring back to noun phrases. But there is the possibility of using the semantic role of the pronoun to find antecedents that are compatible, e.g. as subject or object of the pronoun's sentence's predicate, as pointed out by Johansson et al. (2005). As there is plenty of room for improvement in co-reference and anaphora resolution, this might be a suitable task to evaluate ontologies that encode semantic roles additionally to hierarchical relations.

4 Conclusion

After clarifying the usage of the term *ontology*, a variety of methods have been described how to construct and extend ontologies using unstructured text sources. We have then been looking at approaches that are directly labeled with *ontology learning*, complemented by a consideration of earlier work that has similar goals despite differing terminology. Further, various scenarios for ontology evaluation have been conveyed.

Currently, ontology learning cannot fulfill the promises that its name suggests. As far as prototype-based ontologies are concerned, clustering might yield more or less semantically coherent sets of words, but will not be of great help for carrying out the crucial step from terms to concepts. Taxonomical ontologies can be learnt as far as the relations are explicitly mentioned in the text and extractable by patterns that are scarcely met in real life texts. For circumventing the problem of possible pattern mismatches (i.e. "life is a highway") even more text has to be considered, resulting in very small taxonomies as opposed to the size of the corpus, as pointed out by Brewster et al. (2005).

Especially when comparing the requirements for formal ontologies as formulated by the Semantic Web community and the structures learnable from text as described, one has to state that the 'self-annotating web' will remain a vision for a long time.

But maybe the task is ill-defined. It is beyond doubt that modeling semantics will carry natural language processing further, as it has reached a state where further improvement of systems would in fact need rather more language understanding than more rules or more training examples. It is an open question, however, whether formal specifications are the only way to reach the goal, or whether the manual approach of hand-coding semantics will be outperformed by inconsistent, statistical black-box methods again.

5 Acknowledgements

The author would like to thank Gerhard Heyer and Christer Johansson for useful comments. This work was partially carried out at MULTILINGUA, University of Bergen, supported by the European Commission under the Marie Curie actions.

References

Agichtein, E. and L. Gravano (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*.

Agirre, E., O. Ansa, E. Hovy, and D. Martinez (2000). Enriching very large ontologies using the WWW. In *Proceedings of the ECAI 2000 Workshop on Ontology Learning*, Berlin, Germany.

Alfonseca, E. and S. Manandhar (2002). Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, Berlin, pp. 1–7. Springer.

Berland, M. and E. Charniak (1999). Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*.

Biemann, C., S. Bordag, and U. Quasthoff (2004). Automatic acquisition of paradigmatic relations using iterated co-occurrences. In *Proceedings of LREC 2004*, Lisboa, Portugal.

Biemann, C. and R. Osswald (2005). Automatische Erweiterung eines semantikbasierten Lexikons durch Bootstrapping auf großen Korpora. In B. Fisseni, H.-C. Schmitz, B. Schröder, and P. Wagner (Eds.), *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005, Universität Bonn*, Frankfurt am Main. Peter Lang.

Biemann, C., S.-I. Shin, and K.-S. Choi (2004). Semiautomatic extension of CoreNet using a bootstrapping mechanism on corpus-based co-occurrences. In *Proceedings of the 20th Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, pp. 1227–1232.

Bisson, G., C. Nédellec, and L. Cañamero (2000). Designing clustering methods for ontology building – the Mo'K workbench. In *Proceedings of the ECAI 2000 Workshop on Ontology Learning*, Berlin, Germany.

Bontcheva, K., V. Tablan, D. Maynard, and H. Cunningham (2004). Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering 10*(3/4), 349–373.

Brewster, C., H. Alani, S. Dasmahapatra, and Y. Wilks (2004). Data driven ontology evaluation. In *Proceedings of LREC 2004*, Lisboa, Portugal.

Brewster, C., J. Iria, F. Ciravegna, and Y. Wilks (2005). The Ontology: Chimaera or Pegasus. In *Proceedings of the Dagstuhl Seminar Machine Learning for the Semantic Web*, Dagstuhl, Germany.

Brin, S. (1998). Extracting patterns and relations from the World Wide Web. In *WebDB Workshop at the 6th International Conference on Extending Database Technology (EDBT'98)*.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pp. 120–126.

Chinchor, N. A. (1998). Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Choi, K.-S. and H.-S. Bae (2004). Procedures and problems in Korean-Chinese-Japanese WordNet with shared semantic hierarchy. In *Global WordNet Conference*, Brno, Czech Republic.

Cimiano, P., A. Hotho, and S. Staab (2004). Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pp. 435–443.

Cimiano, P. and S. Staab (2004). Learning by googling. *SIGKDD Explorations 6*(2), 24–34.

Cimiano, P. and S. Staab (2005). Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods (OntoML 05)*, Bonn, Germany.

Dagan, I., F. C. N. Pereira, and L. Lee (1994). Similarity-based estimation of word co-occurrence probabilities. In *Meeting of the Association for Computational Linguistics*, pp. 272–278.

de Saussure, F. (1916). *Cours de linguistique générale*. Paris: Payot.

Deerwester, S. C., S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science 41*(6), 391–407.

Ding, Y. and S. Foo (2002). Ontology research and development: Part 1 – A review of ontology generation. *Journal of Information Science 28*(2), 123–136.

Dornseiff, F. (2004). *Der deutsche Wortschatz nach Sachgruppen. 8., völlig neu bearb. u. mit einem vollständigen alphabetischen Zugriffsregister versehene Aufl. von Uwe Quasthoff*. Berlin, New York: Walter de Gruyter.

Eco, U. (1977). *A Theory of Semiotics*. London: The Macmillan Press.

Faure, D. and C. Nédellec (1998). ASIUM: Learning subcategorization frames and restrictions of selection. In Y. Kodratoff (Ed.), *Proceedings of 10th Conference on Machine Learning (ECML 98): Workshop on Text Mining*, Chemnitz, Germany.

Fleischman, M. and E. Hovy (2002). Fine grained classification of named entities. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Gale, W. A., K. W. Church, and D. Yarowsky (1993). A method for disambiguating word senses in a large corpus. *Computers and the Humanities 26*, 415–439.

Gliozzo, A., C. Giuliano, and C. Strapparava (2005). Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, USA, pp. 403–410.

Gómez-Pérez, A. and D. Manzano-Macho (2003). A survey of ontology learning methods and techniques. Deliverable 1.5, OntoWeb Project.

Grishman, R. (1997). Information extraction: Techniques and challenges. In *SCIE*, pp. 10–27.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition 5*(2), 199–220.

Harris, Z. S. (1968). *Mathematical Structures of Language*. New York: Interscience Publishers John Wiley & Sons.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 1992)*, Volume 2, Nantes, France, pp. 539–545.

Heinrich, G., J. Kindermann, C. Lauth, G. Paaß, and J. Sanchez-Monzon (2005). Investigating word correlation at different scopes – a latent topic approach. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by using Machine Learning (OntoML 05)*, Bonn, Germany.

Helbig, H. (2001). *Die semantische Struktur natürlicher Sprache*. Heidelberg: Springer.

Heyer, G., U. Quasthoff, and T. Wittig (2005). *Wissensrohstoff Text*. Bochum: W3L-Verlag.

Heyer, G. and H. F. Witschel (2005). Terminology and metadata – on how to efficiently build an ontology. *TermNet News – Newsletter of International Cooperation in Terminology 87*.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Meeting of the Association for Computational Linguistics*, pp. 268–275.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, Sweden, pp. 289–296.

Hoste, V. (2005). *Optimization Issues in Machine Learning of Coreference Resolution*. Ph. D. thesis, University of Antwerp, Belgium.

Husserl, E. (1975). *Logische Untersuchungen 1: Prolegomena zur reinen Logik*. Husserliana 18 (edited by E. Holenstein). Den Haag.

Johansson, C., A. Nøklestad, and C. Biemann (2005). Why the monkey ate the banana. In *Proceedings of the Workshop on Anaphora Resolution (WAR)*, Mjølfjell, Norway.

Keller, F., M. Lapata, and O. Ourioupina (2002). Using the web to overcome data sparseness. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, USA, pp. 230–237.

Kilgarriff, A. (1998). SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proceedings of LREC 1998*, Granada, Spain, pp. 581–588.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM 38*(11), 33–38.

Leopold, E. (2005). On semantic spaces. *LDV-Forum (Special Issue on Text Mining) 20*(1), 63–86.

Leveling, J. and S. Hartrumpf (2005). University of Hagen at CLEF 2004: Indexing and translating concepts for the GIRT task. In C. Peters, P. Clough, J. Gonzalo, G. J. F. Jones, M. Kluck, and B. Magnini (Eds.), *CLEF 2005*, pp. 271–282. Berlin: Springer.

Maedche, A. and S. Staab (2002). Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW-2002)*, Berlin, pp. 251–263. Springer.

Maedche, A. and S. Staab (2004). Ontology learning. In S. Staab (Ed.), *Handbook on Ontologies*, pp. 173–190. Springer.

Miller, G. A. (1990). WordNet – an on-line lexical database. *International Journal of Lexicography 3*(4), 235–244.

Omelayenko, B. (2001). Learning of ontologies for the web: the analysis of existent approaches. In *Proceedings of the International Workshop on Web Dynamics*.

Paaß, G., J. Kindermann, and E. Leopold (2004). Learning prototype ontologies by hierarchical latent semantic analysis. In *Knowledge Discovery and Ontologies (KDO-2004)*, Pisa, Italy.

Paşca, M. (2005). Finding instance names and alternative glosses on the Web: WordNet reloaded. In *Proceedings of Computational Linguistics and Intelligent Text Processing: 6th International Conference (CICLing 2005)*, LNCS 3406, Mexico City, Mexico, 2005, pp. 280–292.

Pantel, P., D. Ravichandran, and E. Hovy (2004). Towards terascale knowledge acquisition. In *Proceedings of the 20th Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Pease, A. and I. Niles (2002). IEEE standard upper ontology: a progress report. *Knowledge Engineering Review, Special Issue on Ontologies and Agents 17*(1), 65–70.

Pereira, F., N. Tishby, and L. Lee (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190.

Quine, W. V. (1969). *Ontological Relativity and Other Essays*. New York: Columbia University Press.

Rapp, R. (2002). The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Reuters Corpus (2000). Volume 1, English language, 1996-08-20 to 1997-08-19, release date 2000-11-03, format version 1. `http://about.reuters.com/researchstandards/corpus`.

Rieger, B. B. (1981). Feasible fuzzy semantics. On some problems of how to handle word meaning empirically. In H. Eikmeyer and H. Rieser (Eds.), *Words, Worlds, and Contexts. New Approaches in Word Semantics (Research in Text Theory 6)*, pp. 193–209. Berlin/New York: de Gruyter.

Riloff, E. and R. Jones (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI-99*, pp. 474–479.

Riloff, E. and J. Shepherd (1997). A corpus-based approach for building semantic lexicons. In C. Cardie and R. Weischedel (Eds.), *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)*, Somerset, New Jersey, USA, pp. 117–124. Association for Computational Linguistics.

Ritter, J. and K. Gründer (Eds.) (1995). *Historisches Wörterbuch der Philosophie*. Basel/Stuttgart: Schwabe.

Roark, B. and E. Charniak (1998). Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the ACL*, pp. 1110–1116.

Roget, P. (1852). Roget's thesaurus of english words and phrases. In *Longman*, London.

Ruge, G. (1992). Experiment on linguistically-based term associations. *Information Processing and Management 28*(3), 317–332.

Salton, G., A. Wong, and C. S. Yang (1975). A vector space model for automatic indexing. *Communications of the ACM 18*(11), 613–620.

Sánchez, D. and A. Moreno (2005). Web-scale taxonomy learning. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by using Machine Learning (OntoML 05)*, Bonn, Germany.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics 24*(1), 97–123.

Smith, B. (2004). Ontology. In L. Floridi (Ed.), *The Blackwell Guide to Philosophy of Computing and Information*. Blackwell: Malden.

Sowa, J. F. (2003). Ontology. `http://www.jfsowa.com/ontology/` (last changed 2003).

Staab, S., C. Braun, A. Düsterhöft, A. Heuer, M. Klettke, S. Melzig, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, and B. Wrenger (1999). GETESS – Searching the web exploiting german texts. In *CIA'99: Proceedings of the Third International Workshop on Cooperative Information Agents III*, London, UK, pp. 113–124. Springer.

Stamou, S. and D. Christodoulakis (2005). Retrieval efficiency of normalized query expansion. In *Proceedings of Computational Linguistics and Intelligent Text Processing: 6th International Conference (CICLing 2005)*, LNCS 3406, Mexico City, Mexico, pp. 593–596.

Thelen, M. and E. Riloff (2002). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, USA.

Vossen, P. (1997). EuroWordNet: A multilingual database for information retrieval. In *Proceedings of the DELOS workshop on Cross-language Information Retrieval, March 5-7, 1997*, Zürich, Switzerland.

Wagner, A. (2000). Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In *Proceedings of the ECAI 2000 Workshop on Ontology Learning*, Berlin, Germany.

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT-NAACL 2003: Main Proceedings*, pp. 276–283.

Witschel, H. F. (2005). Using decision trees and text mining techniques for extending taxonomies. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by using Machine Learning (OntoML 05)*, Bonn, Germany.

Autorenverzeichnis

Chris Biemann
Fakultät für Mathematik und Informatik
Institut für Informatik
Universität Leipzig
Augustusplatz 10-11
04109 Leipzig
biem@informatik.uni-leipzig.de

Thierry Declerck
DFKI GmbH
Language Technology Lab
Stuhlsatzenhausweg 3
66123 Saarbrücken
declerck@dfki.de

Peter Grzybek
Institut für Slawistik
Karl-Franzens-Universität Graz
Merangasse 70
A-8010 Graz
peter.grzybek@uni-graz.at

Ulrich Heid
Institut für Maschinelle
Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
70174 Stuttgart
ulrich.heid@ims.uni-stuttgart.de

Emmerich Kelih
Institut für Slawistik
Karl-Franzens-Universität Graz
Merangasse 70
A-8010 Graz
emmerich.kelih@uni-graz.at

Reinhard Köhler
Linguistische Datenverarbeitung
Universität Trier
54286 Trier
koehler@uni-trier.de

Georg Rehm
http://georg-re.hm
georg.rehm@gmail.com

Thorsten Trippel
Fakultät für Linguistik und
Literaturwissenschaft
Universität Bielefeld
Postfach 100131
33501 Bielefeld
thorsten.trippel@uni-bielefeld.de