



Visualisierung reich annotierter linguistischer Daten

Magisterarbeit
zur Erlangung des akademischen Grades Magistra Artium (M.A.)
im Fach Germanistische Linguistik

Humboldt-Universität zu Berlin
Philosophische Fakultät II
Institut für deutsche Sprache und Linguistik

eingereicht von: Katja Futterleib
geb. am 13.11.1972
in Hamburg

Wissenschaftliche Betreuer:
Prof. Dr. Pompino-Marschall (Erstbetreuung)
Prof. Dr. Karin Donhauser (Zweitbetreuung)



*„The rising complexity of current data sets requires new techniques
to support users to handle their visual analysis“*

(Nocke, 29-31 March 2004)

Inhaltsverzeichnis

I	Einleitung	6
1	Motivation	6
2	Problemstellung	6
3	Methode	8
II	Linguistische Datenbanken und Annotation	9
1	Begriffe	9
2	Datenbank des SFB 632 Informationsstruktur (ANNIS)	11
3	Potsdamer Commentary Corpus (PCC)	13
4	Visualisierungen im Explorationsprozess	14
III	Allgemeine Anforderungen	16
1	Wissenschaftliche Visualisierung qualitativer Daten	16
2	Direkte Manipulation	20
3	Web-Interface	21
IV	Spezifizierung der Anforderungen/ Requirements Specification	23
1	Nutzergruppe „Linguisten“ (SFB 632 + PCC)	23
2	Zwei linguistische Fragestellungen (Szenarien)	25
	1 Szenario 1: Informationsstruktur und Verbstellung im Althochdeutschen ...	25
	2 Szenario 2: Modelle der Textstruktur	27
3	Gegenstandseingrenzung	28
V	Evaluation/Bewertung ausgewählter Visualisierungswerkzeuge	29
1	Auswahlkriterien und Bewertungsrahmen	29
2	ANNIS	30
3	EXMARALDA	35
4	MMAX 2	38
5	RSTTool	40
6	TIGERSearch	42
VI	Möglichkeiten der Visualisierung von Text und Annotationen	44
	1 Intralineare Visualisierung von Annotationen	46

	2	Interlineare Visualisierung von Annotationen	49
	3	Extralineare Visualisierung von Annotationen	50
	4	Superlineare Visualisierung von Annotationen	51
	5	Zusammenfassung	51
VII		Konzept für die Visualisierung reich annotierter linguistischer Daten	53
	1	Guidelines	53
		1 Farben	53
		2 Maßangaben	55
	2	Kontext: User Interface zur Exploration linguistischer Daten	55
	3	kombinierte vs. integrierte Darstellung	56
	4	Basisvisualisierung: intralineare Darstellung	58
	5	Basisvisualisierung: interlineare Darstellung (Partitur)	59
		1 Visualisierung von hierarchischen Annotationen (Syntax)	66
		2 Interlineare Darstellung von Position in Ergebnislisten	70
	6	Basisvisualisierung: extralineare Darstellung (Baum)	71
		1 Extralineare Darstellung von Bäumen: RSTTool (horizontal)	71
		2 Extralineare Darstellung von Bäumen: RST vertikal	71
	7	Basisvisualisierung: superlineare Darstellung	73
	8	Komplexe Darstellung reich annotierter linguistischer Daten	75
VIII		Konzept: technische Realisierung	80
IX		Zusammenfassung	85
X		Literaturverzeichnis	87
XI		Anhang	90

Kurzbeschreibung

In der germanistischen Linguistik wächst die Bedeutung der Datenexploration auf der Basis von digitalen Datenbeständen. Für eine effektive Exploration der Daten sind geeignete Visualisierungen erforderlich, die der zunehmenden Heterogenität der linguistischen Daten, ihrer Nutzer und Fragestellungen gerecht werden. Spezifische Visualisierungen (Baumdarstellung, Partitur) existieren bisher in lokal zu installierenden Tools. Für projektübergreifende Analysen ist eine Verfügbarkeit der Daten über das Internet jedoch von Vorteil. Im Rahmen dieser Arbeit werden Möglichkeiten der integrierten und kombinierten Darstellung reich annotierter linguistischer Daten am Fall der Datenbank ANNIS des SFB 632 „Informationsstruktur“ und des Potsdamer Commentary Corpus (PCC) untersucht und Möglichkeiten der technischen Realisierung aufgezeigt.

Einleitung

1 Motivation

In der germanistischen Linguistik (wie in vielen anderen Fachbereichen) ist die Bedeutung der Datenexploration auf der Basis von digitalen Datenbeständen in den letzten Jahrzehnten kontinuierlich gewachsen.

Linguistische Datenbanken (Korpora) beinhalten Daten für weit reichende linguistische Analysen linguistischer Fragestellungen. Diese Daten liegen jedoch zunächst in einem Format vor, das für die direkte Exploration durch den Nutzer eher ungeeignet ist. Eine effektive Auswertung ist dagegen gebunden an eine expressive und effektive Visualisierung der Daten, die die Analyse, das Verständnis und die Kommunikation von Modellen, Konzepten und Daten erleichtert.

Verschiedene Visualisierungen sind bereits im linguistischen Kontext entwickelt worden. Sie sind integriert in lokale (z.B. EXMARaLDA, TIGERSearch) und webbasierte Interfaces (z.B. KWIC-Liste in COSMAS II) und unterstützen den Nutzer sowohl beim Erstellen von Annotationen als auch bei der Exploration der Daten. Im Vordergrund stand dabei bisher die Expressivität der Visualisierungen, nicht immer jedoch auch die Frage der Effektivität. Dass jedoch nicht jede expressive Visualisierung automatisch auch effektiv ist, zeigt sich deutlich am Fall des aktuellen Web-Interfaces ANNIS des Sonderforschungsbereiches 632 „Informationsstruktur“¹ der Humboldt-Universität zu Berlin und der Universität Potsdam, das von einem Teil der Nutzer u.a. wegen kontraintuitiver Visualisierungen – aber auch wegen unzureichender Nutzerfreundlichkeit des (allerdings prototypischen!) Interfaces – nicht akzeptiert wird.

Diese Arbeit ist motiviert durch den Wunsch, die Explorationsmöglichkeiten der vielfältigen linguistischen Daten in der Datenbank ANNIS mit Hilfe intuitiv erfassbarer Visualisierungen zu verbessern und damit die Nutzerakzeptanz zu erhöhen.

2 Problemstellung

Die Datenbank des SFB 632 ist gekennzeichnet durch die Heterogenität ihrer Daten, ihrer Nutzer und der Fragestellungen, hinsichtlich derer die Nutzer die Daten analysieren möchten.

Vorhandene lokale und webbasierte Interfaces unterstützen meist nur ganz spezifische Daten: Die Oberfläche von TIGERSearch unterstützt die Darstellung syntaktisch anno-

tierter Daten, das RST-Tool visualisiert Daten, die nach der Rhetorical Structure Theory annotiert sind, und EXMARaLDA stellt in einer Partiturdarstellung "normale" Attribut-Wert-Paare dar.

Sofern die Daten einer Datenbank nur auf einer einzigen linguistischen Ebene beschrieben wurden, ist diese Spezifizierung der Darstellung nicht weiter problematisch. Die Daten können einfach mit dem entsprechenden Tool visualisiert werden – vorausgesetzt, die Daten liegen in einem Format vor, das von dem Tool interpretiert werden kann. Schwierig wird es hingegen, wenn die Daten auf mehr als einer linguistischen Ebene annotiert sind und diese Annotationen sich hinsichtlich ihrer Datenstruktur und damit auch ihrer typischen Visualisierungsform unterscheiden (= so genannte *reich annotierte* Daten oder *heterogene Mehrebenenannotation*). Da ein Tool in der Regel nur eine einzige Visualisierungsform zur Verfügung stellt, können die Daten nur in verschiedenen Tools parallel oder aber nacheinander untersucht werden. Die spezifisch visualisierten Daten aus verschiedenen Tools direkt zueinander in Beziehung zu setzen, wird dadurch erschwert.

Die Nutzer der Datenbank ANNIS sind vielfältig. Sie unterscheidet u.a. der theoretische Hintergrund (Phonetiker/Phonologen, Sprachhistoriker, Computerlinguisten), die Fragestellung, die mit einer Datenbankabfrage überprüft werden soll und die Nutzungskompetenz von Datenbanken und damit ihre Erwartung an eine Nutzerschnittstelle.

Diese Arbeit geht der Frage nach, auf welche Art sich heterogene linguistische Daten so darstellen lassen, dass heterogene Nutzer mit heterogenen Fragestellungen bei der (qualitativen) Exploration der Daten unterstützt werden.

Ziel ist die Konzeption und Teilimplementation von webbasierten Visualisierungen für reich annotierte linguistische Daten, ausgehend von ausgewählten Daten, Nutzern und Fragestellungen des SFB 632 „Informationsstruktur“ und des Potsdamer Commentary Corpus (PCC). Es wird eine flexible und adaptierbare visuelle Analyse-Umgebung angestrebt, die verschiedene Kontexte und nutzerspezifische Ziele berücksichtigt. Die Implementierung kann im Rahmen dieser Arbeit nicht vollständig erfolgen, und muss somit auf ausgewählte Aspekte beschränkt werden. Da sich die Visualisierungen an den Nutzerbedürfnissen orientieren sollen, ist es sinnvoll, den Prozess der Visualisierungen nah am Nutzer zu beginnen. Das heißt, dass im Vordergrund effektive Visualisierungen stehen die dann in einem zweiten Schritt (und nicht mehr vollständig im Rahmen dieser Arbeit) implementiert werden können.

3 Methode

Ausgangspunkt für die Konzipierung von Visualisierungen linguistischer Daten, ist eine Anforderungsspezifizierung. Für eine Datenanalyse werden zum einen die Encoding Standards des SFB 632 für die Annotierung der Daten herangezogen sowie reale Daten aus den Projekten B4 und D1/PCC betrachtet. Für eine Nutzeranalyse werden Interviews mit Vertretern verschiedener SFB-Projekte geführt. Für eine Analyse von Fragestellungen und Explorationsstrategien werden ebenfalls Interviews, speziell mit Vertretern der Projekte B4 und D1/PCC geführt. Auf der Grundlage der definierten Anforderungen bewerte ich im Anschluss spezifische Visualisierungen ausgewählter Software-Tools und definiere in einem nächsten Schritt die visuellen Variablen, die für eine Visualisierung reich annotierter linguistischer Daten grundlegend sind.

Mittels dieser Variablen werden Visualisierungen konzipiert und soweit umgesetzt, wie es für eine Demonstration und weitere ausführlichere Nutzertests geeignet ist.

Linguistische Datenbanken und Annotation

Linguistische Datenbanken werden in verschiedenen linguistischen Bereichen genutzt, um theoretische Hypothesen an konkretem Sprachmaterial empirisch zu überprüfen. Die historische Linguistik überprüft z.B. den Gebrauch eines Wortes in einem bestimmten Zeitraum (vgl. Kennedy 1998).

Die Psycholinguistik beschäftigt sich mit der Frage, ob die Häufigkeit eines Wortes einen Einfluss auf die Geschwindigkeit hat, in der ein Wort erkannt wird. Und in der Phonetik/Phonetik möchte man untersuchen, welche Varianten es für die Aussprache von Fremdwörtern gibt. Um solche Fragen beantworten zu können, werden Sprachdaten (nach mehr oder weniger definierten Kriterien) erhoben und ausgewertet. Eine automatische Auswertung nach spezifischen Parametern kann dabei besser erfolgen, wenn die erhobenen Rohdaten zuvor mit expliziten linguistischen Informationen versehen wurden (vgl. MacEnery & Wilson 2001: 32), d.h. wenn sie **annotiert** worden sind.

Wenn Textdaten erst einmal systematisch beschrieben wurden, können sie nach den zugrunde liegenden Beschreibungskategorien (und den jeweiligen Werten) durchsucht werden. So wird es nicht nur möglich, Belegstellen für bestimmte sprachliche Phänomene zu finden, sondern auch, Beschreibungskategorien (und ihre Werte) zueinander in Beziehung zu setzen, neue Hypothesen über sprachliche Phänomene zu formulieren und diese (qualitativ und quantitativ) zu überprüfen.

1 Begriffe

Im Folgenden werden einige grundlegende Begriffe definiert, auf die ich mich im Rahmen dieser Arbeit beziehen werde.

Die erhobenen Daten werden als **Rohdaten** bezeichnet. Diese können als unterschiedliche Medien vorliegen: als Audio- oder Videodaten ebenso wie als Text. Von Audio- und Videodaten werden in der Regel Transkripte angefertigt, denen dann in einem zweiten Schritt die Beschreibungen zugewiesen werden. Die Textversionen, ob als Rohdaten oder als (orthographische) Transkripte, bezeichne ich im Folgenden als **Primärdaten**².

Die zugewiesenen Beschreibungen können sich entweder auf den Text als ganzen beziehen – dann werden sie als **Metadaten** bezeichnet (dazu gehören z.B. die Anzahl der Sprecher oder das Datum der Erhebung) – oder aber sie beziehen sich als linguistische Informationen auf Einheiten des Textes, dann werden sie **Annotationen** genannt. Das Beschreibungs-

2 Im Gegensatz zu Dipper et al. (2004), die die Primärdaten gleichsetzen mit den Rohdaten.

inventar kann dabei aus verschiedenen linguistischen Bereichen stammen, z.B. aus den Bereichen Phonologie oder Informationsstruktur. Komplexere linguistische Ebenen werden für die Formulierung von Beschreibungseinheiten in Unterebenen aufgeteilt: Syntax wird beispielsweise aufgeteilt in die Ebenen Satzfunktion und Konstituentenkategorie (vgl. Dipper et al. 2004). Eine einzelne Ebene wird als **Annotationsebene** definiert (vgl. ebd.).

Eine Annotation lässt sich über folgende Variablen beschreiben:

- (1) das Segment, auf das sich die Annotation bezieht; das Segment kann auf der Textebene (Tokenebene/ unterhalb der Tokenebene/ oberhalb der Tokenebene) oder hinsichtlich der Zeitskala (z.B. 10 msec) definiert sein; das Segment kann kontinuierlich oder diskontinuierlich z.B. gesplittete NP) sein; es kann sich um ein Segment des Primärtextes handeln oder um ein Annotationssegment einer anderen Beschreibungsebene;
- (2) das Attribut, also die Eigenschaft, hinsichtlich derer das Segment beschrieben wird;
- (3) den (manuell oder automatisch) tatsächlich zugewiesenen Wert aus dieser Wertemenge.

Ein Attribut-Wert-Paar wird auf allgemeiner Ebene als **Tag** bezeichnet, z.B. „pos=NN“ (vgl. ebd.); eine Annotation kann als Instanz eines Tags betrachtet werden. Ein **Tagset** ist „ein Set aus Attribut-Wert-Paaren, die für eine bestimmte linguistische Ebene verfügbar sind“ (ebd.). Dabei gibt es Tagsets, die nur ein einziges Attribut beinhalten (z.B. pos im Tagset STTS³) und solche, die mehrere Attribute (und deren mögliche Werte) beinhalten (z.B. Syntax, beschrieben über die syntaktische Kategorie und die grammatische Funktion) (vgl. ebd.).

Reich annotierte linguistische Daten nenne ich im Folgenden solche Daten, die (a) auf mehreren Ebenen annotiert (*multi-level annotation*⁴, *Mehrebenenannotation*) oder hinsichtlich einer Ebene mehrfach annotiert sind (*multiple annotation*) und die sich (b) hinsichtlich der internen Struktur ihrer Annotationen unterscheiden (*heterogene Annotation*).

(a) Bei den *mehrfach annotierten* Daten handelt es sich um Daten, die auf mehreren linguistischen Ebenen annotiert sind (z.B. Morphologie und Informationsstruktur), oder die auf

³ STTS
⁴(s. MMAX)

mehreren Annotationsebenen innerhalb einer linguistischen Ebene annotiert sind (z.B. Syntax: grammatische Funktion und Konstituentenstruktur), oder die hinsichtlich einer Ebene mehrfach annotiert sind (*multiple annotation*) (z.B. konkurrierende Annotationen zur Informationsstruktur oder Annotationen nach verschiedenen Tagsets).

(b) *Heterogene Annotationen* sind gekennzeichnet durch die Unterschiede der internen Struktur der Annotationen, z.B. einfaches Attribut-Wert-Paar vs. hierarchische Struktur vs. Koreferenz.

Die Unterschiede der internen Struktur seien hier kurz dargestellt: Bei einfachen Annotationen (in Abgrenzung zu den anderen Formen „einfaches“ Attribut-Wert-Paar oder kurz schlicht „Attribut-Wert-Paar“ genannt) bezieht sich das Attribut-Wert-Paar auf ein Textsegment des Primärtextes.

Bei der Annotation von hierarchischer Struktur stehen mehrere Attribut-Wert-Paare in hierarchischer Beziehung; dabei bezieht sich ein Attribut-Wert-Paar nicht nur auf ein Textsegment des Primärtextes, sondern auch auf ein Segment einer anderen Ebene, das hinsichtlich des gleichen Attributes annotiert ist.

Bei der Annotation von Koreferenz⁵ bezieht sich ein Attribut-Wert-Paar nicht nur auf ein, sondern auf zwei oder mehrere (voneinander getrennte) Textsegmente des Primärtextes; das Ausgangssegment und das Zielsegment.

2 Datenbank des SFB 632 Informationsstruktur (ANNIS)

Der Sonderforschungsbereich 632 „Informationsstruktur“⁶ ist ein Gemeinschaftsprojekt von 13 Einzelprojekten an der Humboldt-Universität zu Berlin und der Universität Potsdam. Er wurde im Jahr 2003 eingerichtet, um das Phänomen *Informationsstruktur*⁷ ausgehend von unterschiedlichen Sprachen, Forschungshintergründen und -fragestellungen grundlegend zu erforschen. Dementsprechend weit gefächert ist die Auswahl an vertretenen linguistischen Bereichen (Phonologie/Phonetik, Diachronie, Computerlinguistik, Psycholinguistik, Semantik, etc.) sowie auch an betrachteten Sprachen (neben dem

5 Koreferenz bezeichnet „in der Textlinguistik eine textinterne Relation zwischen zwei oder mehreren Textelementen, die auf dasselbe Objekt verweisen“ (Lexikon Sprache: Koreferenz, S. 1. Digitale Bibliothek Band 34: Metzler Lexikon Sprache, S. 5144 (vgl. MLSpr, S. 383) (c) J.B. Metzler Verlag).

6 <http://141.89.100.102/sfb-is/>

7 „Unter Informationsstruktur verstehen wir die Strukturierung von sprachlicher Information, typischerweise zum Zweck der Optimierung des Informationstransfers im Diskurs. Die zugrunde liegende Vorstellung ist, dass dieselbe Information je nach Hintergrund und Ziel des Diskurses verschieden aufbereitet und unterteilt werden muss. Dabei sind drei Aspekte von besonderem theoretischem Interesse: Das Zusammenspiel der relevanten formalen Ebenen (Phonetik, Phonologie, Morphologie, Syntax, Semantik, die Wahl der lexikalischen Mittel, der Aufbau von Texten), die allgemeine kognitive Verarbeitung der Informationsstruktur, und eine sprachvergleichende Typologie der informationsstrukturellen Ausdrucksmittel“. <http://www.sfb632.uni-potsdam.de/main.php?lang=de> (aus der Projektbeschreibung)

Deutschen/Germanischen z.B. die afrikanischen Gur- und Kwasprachen oder Japanisch).

Die Datenbank ANNIS (ANNotierte InformationsStruktur) wird im Rahmen des SFB 632 Informationsstruktur (im Projekt D1) entwickelt, mit dem Ziel, die Nutzer bei der Auswertung ihrer vielfältigen Projektdaten zu unterstützen und die Ergebnisse nach außen zu präsentieren. Sie umfasst gegenwärtig insgesamt **1952 Dokumente**, bestehend aus Projektdaten der Projekte B4 („Die Rolle der Informationsstruktur bei der Herausbildung von Wortstellungsregularitäten des Germanischen“)⁸, D2 („Typologie der Informationsstruktur“) und B2 („Fokussierung in den tschadischen Sprachen“) sowie aus Daten des Potsdamer Commentary Corpus (s.u.).

Die Daten in ANNIS sind vielfältig. Es sind verschiedene Textsorten (mündlich/schriftlich; monologisch/dialogisch), verschiedene Medientypen (Audio, Video, Bild, Text), verschiedene Sprachen und Sprachstufen repräsentiert. Zudem sind die Daten auf den unterschiedlichsten linguistischen Ebenen beschrieben.

Die Daten sind hinsichtlich eines eigens entwickelten allgemeinen SFB 632 *Encoding Standards*⁹ annotiert worden und liegen in digitaler Form in einem so genannten XML-basierten Standoff-Format vor (Dipper 2005), das sich durch eine Trennung von Primärtext und Annotationen auszeichnet. Dieses Verfahren ermöglicht es, mehrere Annotationsebenen getrennt voneinander zu behandeln, was bei solchen Annotationsebenen von Vorteil ist, deren Bezugssegmente überlappen, bzw. solchen, die auf der komplexeren Baum-Ebene konfligieren. Konkurrierende Annotationen werden hier in getrennten Dateien gespeichert, um überlappende Ebenen zu trennen und verschiedene Hierarchien zu repräsentieren. Aus Gründen der Performanz wird jedoch für die Visualisierung eine Inline-Version der Daten generiert, in der die Daten innerhalb einer Datei vorliegen. Wie das Standoff-Format so ist auch die Inline-Version ein XML-Format (Extensible Markup Language). Aus diesem wird über einen Transformationsprozess via XSLT¹⁰ (XSL Transformations) das gewünschte Ausgabeformat für die Darstellung im Browser erstellt. Das bisherige Interface (prototypisch als JavaApplet entwickelt) unterstützt noch keine spezifische Visualisierung heterogener Daten, wie sie für eine geeignete Exploration der SFB-Daten sinnvoll wäre. Das ist einer der Gründe, aus denen die Datenbank noch nicht von allen empirisch arbeitenden Projekten (Projekte) genutzt wird. Die Projekte verfügen über eine große Anzahl an Daten, die lokal auf den Computern der einzelnen Projekt-

8 1657 Dokumente

9 Der SFB 632 Encoding Standard wird derzeit zur Veröffentlichung vorbereitet.

10 <http://www.w3.org/TR/2007/REC-xslt20-20070123/>

beteiligten gespeichert sind (in einer Größenordnung von hundert bis mehreren tausend). Diese werden jedoch in lokal installierten Tools exploriert und sind nur für den jeweiligen Nutzer verfügbar. Die Daten auch für andere Nutzer explorierbar zu machen (ob SFB intern oder extern), wäre nicht nur vom Aspekt der Nachhaltigkeit von Vorteil, sondern auch für projektübergreifende Datenanalysen interessant.

3 Potsdamer Commentary Corpus (PCC)

Das Potsdamer Commentary Corpus, erstellt an der Universität Potsdam, beinhaltet deutsche Zeitungskommentare, die auf sechs verschiedenen linguistischen Ebenen annotiert sind: (1) Part-of-Speech, (2) Syntax, (3) Diskursstruktur (RST), (4) Diskurskonnectoren¹¹, (5) Koreferenz und (6) Informationsstruktur (vgl. Stede 2004). Das Korpus wird genutzt bzw. wird genutzt werden für Anwendungen im Bereich der Rhetorical Structure, für den Bereich der Textgenerierung und um Modelle der Diskursstruktur weiterzuentwickeln (vgl. ebd). Die verwendeten Annotationssets werden im folgenden Abschnitt kurz beschrieben. Die Informationen darüber sind – sofern nicht anders vermerkt – dem bereits erwähnten Bericht von Manfred Stede (2004) über das PCC entnommen.

Die *Part-of-Speech-Annotationen* basieren auf dem Stuttgart/Tübingen Tag Set¹² (STTS). Dieses Tagset, entwickelt am Institut für maschinelle Sprachverarbeitung an der Universität Stuttgart und dem Seminar für Sprachwissenschaft der Universität Tübingen, besteht aus 54 Tags zur Beschreibung der Ebene Part-of-Speech (vgl. The Stuttgart-Tübingen Tagset (STTS)).

Die Beschreibungen der *syntaktischen* Ebene orientieren sich an den Empfehlungen des TIGER Projektes zur syntaktischen Annotation deutscher Zeitungstexte. Neben der morphologischen Ebene und der Part-of-Speech-Ebene können hier zusätzlich nicht-terminale Knoten eines Syntax-Graphen und die verbindenden Kanten annotiert werden (vgl. Brants et al. 2002).

Die Annotationen der *Diskursstruktur* orientierten sich zunächst an den Diskursrelationen, die 1988 von Mann & Thompson definiert wurden¹³. Da diese Definitionen jedoch keine Mehrfachzuweisungen bei der Beschreibung von Relationen erlaubten, wurden die Annotationen in ein von Reitter & Stede (2003)¹⁴ neu definiertes Format (URML) übertragen,

11 „connectives with scopes“ ({Stede 2004 #174})

12 The Stuttgart-Tübingen Tagset (STTS) <http://www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts.html>,

13 Mann & Thompson (1988). Rhetorical Structure Theory: A Theory of Text Organization. TEXT 8(3):243-281.

14 Reitter & Stede (2003)...

das multiple Zuweisungen ermöglicht und damit auch vergleichende Analysen konkurrierender Annotationen unterstützt. Die Annotation von *Diskurskonnektoren* (z.B. „und“, „obwohl“) und ihrem Skopus (Geltungsbereich) orientiert sich an der Penn Discourse Tree Bank¹⁵. Bei der Annotation von *Koreferenz* wird unterschieden zwischen anaphorischen Beziehungen und Bridging-Beziehungen (=“definite noun phrases picking up an antecedent via world-knowledge“); es werden PCC-eigene Annotationsrichtlinien angewendet. Die Annotation von *Informationsstruktur* schließlich orientiert sich an Annotationsvorschlägen, die 2003 von Michael Götze entwickelt wurden und die Behandlung von konfligierenden und überlappenden Konzepten berücksichtigt, die den Bereich der Informationsstruktur kennzeichnen.

Für die Darstellung der Daten und eine Suche in den Daten greift das Potsdamer Commentary Corpus auf das im SFB 632 entwickelte Webinterface ANNIS zurück.

4 Visualisierungen im Explorationsprozess

„Exploring information collections becomes increasingly difficult as the volume and diversity grows.“ {Shneiderman 2005 #5 : 561}

So unterschiedlich die Nutzer und ihre Fragestellungen auch sind, bei der Datenexploration bleiben sie alle gebunden an die grundlegenden Phasen eines Suchprozesses¹⁶. Visualisierungen der linguistischen Daten können nicht unabhängig von diesen Phasen betrachtet werden, sondern müssen vielmehr auf diese abgestimmt sein, damit die Exploration der Daten wirkungsvoll unterstützt werden kann.

Als **Exploration** bezeichnen Bortz & Döring (2002) „das mehr oder weniger systematische Sammeln von Informationen über einen Untersuchungsgegenstand (...), das die Formulierung von Hypothesen und Theorien vorbereitet“ (Bortz & Döring 2002:358)

Die Rolle der Visualisierung im Suchprozess ist (a) die der Darstellung von Suchergebnissen (unabhängig von der Art der Suchanfrage textbasiert oder visuell) und (b) die des Ausgangspunktes für eine Suchanfrage. Damit (b) realisierbar wird, muss (a) realisiert sein. Das große Ziel ist (b), aber im Rahmen dieser Arbeit beschränke ich mich auf (a).

Für die Realisierung einer Suchanfrage können im Nutzer-Interface verschiedene Techniken genutzt werden – eine Technik kann z.B. rein textbasiert sein, oder über ein

¹⁵ <http://www.cis.upenn.edu/~pdtb>

¹⁶ In der Literatur werden in diesem Kontext auch die Begriffe Information Retrieval, Database Management, information gathering, information seeking, filtering und data mining genannt (um nur einige zu nennen) (vgl. {Shneiderman 2005 #5})

mehr oder weniger komplexes visuelles Interface unterstützt werden (einfache Menülisten, Navigationsbaum oder interaktive Darstellungen). Um in Zukunft eine Suche *visueller* Art zu ermöglichen, ist es jedoch zunächst einmal erforderlich, geeignete Visualisierungen zu finden, mittels derer die Suchergebnisse überhaupt erst einmal grafisch dargestellt werden können.

Gegenstandseingrenzung

Es wird in der Konzeptionsphase zunächst ein allgemeines (grafisches) Grundgerüst zu erstellen sein, das die Abfolge bei der Informations/Datensuche exemplarisch verdeutlicht. In dieses Grundgerüst werden dann die eigentlichen Visualisierungen für reich annotierte linguistische Daten eingebettet, und zwar für jede Phase des Information Retrieval Prozesses. Da im Rahmen dieser Arbeit nicht jede Visualisierung bis ins Detail ausgearbeitet und umgesetzt werden kann, sind dabei Prioritäten zu setzen.

Dies geschieht nach den folgenden Gesichtspunkten:

1. Welche Visualisierung ist grundlegend für die Exploration – in dem Sinne, dass andere Visualisierungen ohne sie nicht sinnvoll sind? Auf diese sei der Schwerpunkt gesetzt.
2. Gibt es bereits Visualisierungen für die jeweilige Phase des Information Retrieval Prozesses?

Dann sei auf diese verwiesen, sofern sie nicht grundlegend nach 1. sind. Wenn sie grundlegend sind, sollte überprüft werden, welche der bestehenden Techniken am sinnvollsten für die Visualisierung reich annotierter linguistischer Daten adaptiert werden kann.

Allgemeine Anforderungen

1 Wissenschaftliche Visualisierung qualitativer Daten

„*The purpose of computation is insight, not numbers*“ (Hamming 1973, zitiert in Card, Mackinlay & Shneiderman:6)

„*The purpose of visualization is insight, not pictures*“ (Card, Mackinlay & Shneiderman:6)

Während früher unter **Visualisierung** der visuelle Wahrnehmungsprozess des Betrachters (Ware 2004:2) verstanden wurde, umfasst der Begriff heute: (a) den technischen Prozess der Abbildung von Daten (Schumann & Müller 2000) bzw. Informationen (b) das Ergebnis des Abbildungsprozesses (Ware 2004:2) und (c) die Nutzung visueller Darstellungen von Daten (Card et al. 1999:6). Visualisierungen mit interaktivem Charakter wird in neuerer Zeit immer größere Bedeutung beigemessen; sie werden als Mittel betrachtet, mit dessen Hilfe Entscheidungsfindungen bei der Hypothesenbildung erleichtert/unterstützt werden. In diesem Sinne definiert Ware eine Visualisierung auch als „cognitive tool“ (Ware 2004:2). Wissenschaftliche Visualisierung ist also nicht gleichzusetzen mit der Produktion mehr oder weniger schöner Bilder. Vielmehr erfüllt sie im wissenschaftlichen Bereich mehrere Funktionen: (a) Sie ist ein grundlegendes Mittel, um Daten, die im Computer gespeichert sind, überhaupt erst für den Menschen wahrnehmbar zu machen. (b) Sie hat einen großen Einfluss darauf, *ob* und *wie* bereitgestellte Daten/Informationen interpretiert werden – und sollte dementsprechend sorgfältig modelliert werden, damit eine effektive Auswertung der Daten ermöglicht wird (Schumann & Müller 2000). (c) Sie trägt wesentlich zur Akzeptanz einer Nutzeroberfläche bei (*joy of use*).

Der Bereich der Visualisierung wird traditionellerweise in zwei große Gruppen eingeteilt, die Informationsvisualisierung (information visualization¹⁷) und die wissenschaftlich-technische Visualisierung (scientific visualization, data visualization). Die Trennung der beiden Bereiche ist jedoch in weiten Teilen artifiziell: die grundlegenden Modelle des Visualisierungsprozesses, perzeptiven Grundlagen und Ziele ähneln sich stark.

In der Disziplin der wissenschaftlichen Visualisierung (Scientific Visualization)¹⁸ versteht man unter dem Begriff der Visualisierung die Darstellung bestimmter Eigenschaften von Daten in Form von Grafiken oder Sequenzen von Grafiken (vgl. Schumann & Müller

¹⁷ auch *information visualisation*

¹⁸ weitere verwandte Gebiete sind u.a. Data Graphics (Bertin, Tufte), Semiology, Statistics (Exploratory Data Analysis), Computer Graphics + Artificial Intelligence, User Interface Development /Interaction Design, Information Retrieval, Knowledge Discovery in Databases (KDD), Knowledge Visualization und Visual Analytics. Die Übergänge zwischen den Disziplinen sind fließend.

2000). Dabei werden den spezifischen Eigenschaften von Daten spezifische visuelle Attribute zugeordnet (*Mapping*). Dies kann in elementarer (z.B. als Farb- oder Textattribute) oder sehr komplexer Form (als 3-dimensionale Repräsentation oder Animation) erfolgen. Das Ziel aller Visualisierungsformen ist die Darstellung von Zusammenhängen, die in den gegebenen Datenmengen vorhanden sind und die ohne eine visuelle Umsetzung kaum oder nur sehr schwer erkennbar wären (vgl. ebd.).

Mit einer Visualisierung soll dem Betrachter ein Modell vermittelt werden, das die in den vorliegenden Daten enthaltenen Informationen, Eigenschaften und Zusammenhänge tatsächlich repräsentiert. Das mentale Modell des Betrachters stellt wiederum eine Abstraktion des visualisierten realen Modells dar und wird sich deshalb nur selten in vollständiger Übereinstimmung mit diesem befinden. Ziel der Visualisierung sollte jedoch sein, die wesentlichen Eigenschaften und Zusammenhänge zu vermitteln (vgl. ebd.).

Als allgemeine Anforderungen an eine wissenschaftliche Visualisierung formulieren

Schumann & Müller (2000) die drei Kriterien *Expressivität*, *Effektivität* und *Angemessenheit*

Expressivität: Unter Expressivität oder Ausdrucksfähigkeit versteht man die Eigenschaft einer Visualisierung, die in den darzustellenden Daten enthaltenen wesentlichen Informationen explizit und unmissverständlich wiederzugeben. Die Expressivität ist somit ein grundlegendes Qualitätsmerkmal einer Visualisierung und bei der Auswahl der Visualisierungstechnik ist zu überprüfen, ob wirklich nur die in den Daten enthaltenen Informationen dargestellt werden, oder ob nicht eine weitere, nicht dem realen Modell entsprechende Interpretationen möglich ist (vgl. ebd.:9).

Effektivität: Eine expressive Visualisierungsform ist nicht gleichzeitig auch effektiv. Wenn die perzeptiven Fähigkeiten des Nutzers und der technische Kontext bei der Gestaltung nicht berücksichtigt wurden, kann die Visualisierung nicht zum Werkzeug bei der Datenexploration werden. Im Gegensatz zum Expressivitätskriterium fließen in die Effektivität also der Benutzer, das Medium, die Zielsetzung und der Kontext der Anwendung mit ein. So ist zum Beispiel eine graphische Darstellung, die quantitative Unterschiede mit verschiedenen Symbolen visualisiert weniger effektiv im Vergleich zu einer Darstellung, aber der die quantitativen Unterschiede mit einem Symbol in verschiedenen Größen verdeutlicht, da sie dem Betrachter auf einen Blick die Unterschiede intuitiv vermittelt (vgl. ebd.:11). Die effektivste Darstellungsmethode muss aber nicht in jedem Fall auch angemessen sein.

Angemessenheit: Als angemessen wird eine Visualisierung bezeichnet, wenn “Aufwand und Nutzen” (S.12) bei der Erstellung der Visualisierung in einem geeigneten Verhältnis stehen. Die Erstellung einer Visualisierung ist üblicherweise mit einem gewissen technischen und personellen Aufwand verbunden, der durch die Bedeutung der in den Daten enthaltenen Informationen gerechtfertigt sein muss. Hierbei wird auch der enge Zusammenhang von Effektivität und Angemessenheit deutlich (vgl. ebd.:12).

Diese drei Kriterien sind ausschlaggebend für die Qualität einer Visualisierung. Das Expressivitätskriterium muss immer erfüllt werden, weil „nur dann eine unverfälschte Vermittlung aller relevanten Aspekte der Daten an den Betrachter garantiert und das Risiko einer Fehlinterpretation und damit falscher Schlußfolgerungen reduziert werden kann“ (ebd.:12). Die Kriterien Effektivität und Angemessenheit jedoch sind ebenso wichtig.

Grundlegend für die Wahl einer Visualisierung sind – etwas konkreter – nicht nur die Art und Struktur der vorliegenden Daten (z.B. Datentyp, Dimension und Struktur des Beobachtungsbereiches) sowie das Ziel der Visualisierung (Überblick, Detailanalyse, Ergebnispräsentation), sondern besonders auch der Betrachter. Hierbei sind sein Vorwissen, seine visuellen Fähigkeiten, seine Vorlieben und die Konventionen, in die der Betrachter eingebunden ist, für die Konzeption der Visualisierung bzw. der Auswahl der Visualisierungsmethode ausschlaggebend. Schließlich werden durch die Eigenschaften des Visualisierungsmediums Grenzen gesetzt (Auflösung, darstellbare Farben, Rechenleistung etc.), die berücksichtigt werden müssen (vgl. ebd.:8f.).

In der Informations- bzw. Datenvisualisierung sind bereits diverse Visualisierungsmethoden vorgestellt und diskutiert worden. Um die Auswahl geeigneter Visualisierungstechniken zu erleichtern, sind von verschiedenen Autoren diverse Klassifizierungen vorgeschlagen worden. Für die Visualisierung von Daten hat Mackinlay (1986) folgendes Schema erstellt, in dem die Effektivität der einzelnen visuellen Variablen in Abhängigkeit vom zu visualisierenden Datentyp dargestellt ist:

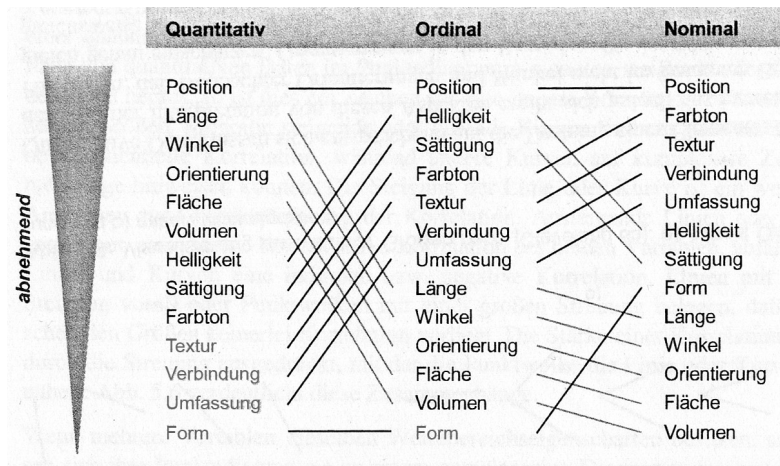


Abbildung 1: Effektivität von visuellen Variablen in Abhängigkeit vom Datentyp

Für nominale Daten ist demnach die visuelle Variable „Positionierung“, gefolgt vom „Farbton“ am effektivsten. Unter visuellen Variablen versteht Schumann (2000:8) „visuelle Attribute, die im Rahmen der Anwendung einer Visualisierungstechnik zur Kodierung von Merkmalsausprägungen der darzustellenden Daten verwendet werden können“.

2 Direkte Manipulation

„Direct manipulation is appealing to novices, is easy to remember for intermittent users, and, with careful design, can be rapid for frequent users“ (Shneiderman & Plaisant 2005:71).

Die direkte Manipulation ist eine Interaktionsmethode die eine große Attraktivität für Nutzer hat. Die folgende Tabelle zeigt ihre Vor- und Nachteile gegenüber den Interaktionstechniken Menüauswahl, Formulare, Kommandosprachen und natürliche Sprache.

Vorteile	Nachteile
direkte Manipulation <ul style="list-style-type: none"> • präsentiert auf visueller Ebene die zugrunde liegenden Konzepte • ermöglicht einfaches Erlernen = niedrige Lernschwelle • ermöglicht einfaches Memorisieren • hilft Fehler zu vermeiden • regt zur Exploration an • gewährt eine hohe Nutzerzufriedenheit, auch bei Anfängern 	<ul style="list-style-type: none"> • evtl. aufwändig zu programmieren • erfordert u.U. Zeigegerät und grafische Oberfläche
Menüauswahl <ul style="list-style-type: none"> • verkürzt die Lernzeit • vermindert die Anzahl der Tastenanschläge • strukturiert die Entscheidungsfindung • erlaubt die Anwendung von Dialog-Management-Tools • erlaubt einfachen Support von Fehlermeldungen 	<ul style="list-style-type: none"> • birgt die Gefahr einer zu großen Anzahl an Menüs • bremst u.U. regelmäßige Nutzer • relativ großer Platzanspruch • erfordert eine schnelle „display rate“
Formulare <ul style="list-style-type: none"> • erleichtern die Dateneingabe • erfordern nur wenig Erklärung • liefern geeignete Unterstützung • erlauben den Gebrauch von Formular-Management-Tools 	<ul style="list-style-type: none"> • nehmen viel Platz in Anspruch
Kommandosprache <ul style="list-style-type: none"> • ist flexibel • spricht sehr fortgeschrittene Nutzer (Experten) an • unterstützt Nutzerinitiative • ermöglicht das Erstellen von nutzerdefinierten Makros 	<ul style="list-style-type: none"> • nur schwache Fehlerbehandlung • erfordert eine umfangreiche Schulung und einen hohen Grad an Erinnerungsleistung
natürliche Sprache <ul style="list-style-type: none"> • das Erlernen einer speziellen Syntax ist nicht erforderlich 	<ul style="list-style-type: none"> • erfordert Dialog zur Auflösung von Ambiguitäten/Unklarheiten • liefert evtl. zu wenig Kontext • erfordert u.U. mehr Tastenanschläge • liefert unvorhersehbare Ergebnisse

Tabelle 1: Vor- und Nachteile der 5 Interaktionsstile nach Shneiderman & Plaisant

Die Anwendung der Technik der direkten Manipulation scheint vor dem Hintergrund der verschiedenen Nutzergruppen im Rahmen dieser Arbeit angebracht.

3 Web-Interface

Die Daten in ANNIS liegen im XML-Format (Extensible Markup Language)¹⁹ vor. Die Daten eines Dokumentes (eines Primärtextes mit allen seinen Annotationen) sind voneinander getrennt in verschiedenen (über XLINK und XPointer verbundenen) Dateien. Um ein Dokument im Web-Browser anzuzeigen, ist es nötig, die Daten in ein Format zu bringen, dass von einem Web-Browser interpretiert werden kann. Verschiedenen Zielformate sind möglich (z.B. HTML, XHTML²⁰, SVG²¹). Die Übersetzung von einem XML-Format in z.B. XHTML kann mittels einer XSLT²²-Umwandlung der XML-Datei geschehen (clientseitig oder – mit Einschränkungen – serverseitig). XSLT hat den Vorteil, dass es in andere Umgebungen portierbar ist und damit leicht nachgenutzt werden kann. Die Gestaltung ergonomischer Nutzeroberflächen beginnt jedoch notwendigerweise immer erst beim Nutzer, und wird erst in einem zweiten (oder dritten) Schritt voll implementiert. Vor einer (mehr oder weniger) finalen Implementierung sollten Nutzer-tests mit der interaktiven Oberfläche stehen. Es ist nicht unwahrscheinlich, dass Erweiterungen oder Modifizierungen der konzipierten Visualisierungen und damit des Zielformates nötig werden. Mögliche Änderungen aber können mit den Webstandards XHTML, CSS²³ und DOM²⁴ voraussichtlich einfacher modelliert werden als mit XSLT. Es ist außerdem zu diskutieren (und zu testen), inwiefern ein umfassender Einsatz von XSLT für Webanwendungen mit hohem Interaktivitätsgrad überhaupt sinnvoll ist. Der serverseitige Einsatz von XSLT ist nicht unbedingt für Interaktionen geeignet, die eine sofortige visuelle Rückkopplung benötigen. Sofortige Rückkopplungen sind aber für die Perzeption von z.B. Farbmarkierungen Voraussetzung. Ein clientseitiger Einsatz von XSLT gilt ebenfalls als eher langsam.

19 Der Standard wurde vom W3C formuliert und hat den Status einer W3C Recommendation. Die aktuelle Spezifikation (Extensible Markup Language (XML) 1.0 (Fourth Edition)) findet sich online unter <http://www.w3.org/TR/2006/REC-xml-20060816> .

20 XHTML (Extensible HyperText Markup Language) Der XHTML-Standard ist ebenfalls eine W3C Recommendation. Die aktuelle Spezifikation (XHTML™ 1.1 - Module-based XHTML) ist online verfügbar unter <http://www.w3.org/TR/2001/REC-xhtml11-20010531>.

21 SVG (Scalable Vector Graphics) Der Standard wurde vom W3C formuliert und hat den Status einer W3C Recommendation. Die aktuelle Spezifikation (Scalable Vector Graphics (SVG) 1.1 Specification (W3C Recommendation 14 January 2003)) findet sich online unter <http://www.w3.org/TR/2003/REC-SVG11-20030114/> .

22 XSLT (XSL Transformations) Der Standard wurde vom W3C formuliert und hat den Status einer W3C Recommendation. Die aktuelle Spezifikation (Version 2.0) wurde am 23.1.2007 veröffentlicht und ist online zu finden unter: <http://www.w3.org/TR/2007/REC-xslt20-20070123/> .

23 CSS (Cascading Style Sheets) Die aktuelle Spezifikation für die CSS-Recommendation wurde vom W3C veröffentlicht. Die aktuelle Spezifikation (Cascading Style Sheets, level 2 CSS2 Specification) findet sich online unter <http://www.w3.org/TR/1998/REC-CSS2-19980512> .

24 DOM (Document Object Model) Der Standard wurde vom W3C formuliert und hat den Status einer W3C Recommendation. Die aktuelle Spezifikation (Document Object Model (DOM) Level 2 Core Specification Version 1.0) findet sich online unter <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113>.

Ich beschränke mich in dieser Arbeit auf die Gestaltung der (interaktiven) Oberfläche mit XHTML, CSS und DOM, da die Entwicklung eines kompletten nutzerfreundlichen interaktiven Web-Interfaces mehr Ressourcen in Anspruch nimmt als der Rahmen einer Masterarbeit bieten könnte. Diese Arbeit versteht sich primär als konzeptioneller Beitrag zur Visualisierung reich annotierter linguistischer Daten. Gleichwohl soll die technische Realisierung nicht völlig außer Acht gelassen werden, da sie (1) einen nicht zu unterschätzenden Einfluss auf die Qualität einer Visualisierung haben kann (Einfluss der Geschwindigkeit einer Systemreaktion auf die Wahrnehmung/Memorierbarkeit von Elementen) und (2) für eine Demonstration der interaktiven Komponenten sinnvoll ist.

Spezifizierung der Anforderungen/ Requirements Specification

1 Nutzergruppe „Linguisten“ (SFB 632 + PCC)

„If you do not understand who your users (or potential users) are, your product is doomed to failure“
(Courage & Baxter 2005:41)

Um das Themenfeld sowie die potenziellen Nutzer und ihre Daten besser einschätzen zu können, wurde eine teilstandardisierte Nutzerbefragung durchgeführt. In einem Fragebogen wurde nach allgemeinen Anforderungen und der Arbeitsumgebung gefragt, in anschließenden Gesprächen konnten projektspezifische Anforderungen erhoben werden. Dabei wurde die Usability-Methode der sogenannten *Focus Group*, in der zeitgleich mehrere potenzielle Nutzer befragt werden (vgl. Courage & Baxter 2005) zugunsten eines Interviews bewusst vermieden, und zwar aus folgenden Gründen: (1) Die Projektbeschreibungen wiesen darauf hin, dass die Nutzer sich hinsichtlich ihrer Fragestellung und ihrem methodischen Ansatz stark unterscheiden. Es war also mit sehr unterschiedlichen Explorationsstrategien und Anforderungen an eine Visualisierung zu rechnen, die sich nicht im Detail in einem Gruppengespräch hätten in Erfahrung bringen lassen. (2) Es sind Fachgebiete vertreten, die traditionellerweise eher wenig technisch orientiert sind (z.B. Sprachgeschichte, Afrikanistik), andere (z.B. Phonetik/Phonologie), die zwar technisch orientiert sind, aber nicht notwendigerweise jeden Tag neuen Programmcode erstellen, und schließlich jene (Computerlinguisten), für die das Programmieren eine alltägliche Routine darstellt. Um vor allem von den „Normalnutzern“ möglichst authentische Aussagen zu erhalten, die sie eventuell im Beisein von „Experten“ nicht äußern würden, wurden dem zeitsparenden Gruppengespräch die zeitintensiveren Einzelinterviews vorgezogen.

Im Folgenden sollen die wichtigsten Ergebnisse der Nutzerbefragungen kurz vorgestellt werden:

- 1) Die Partitur als Visualisierungsform wurde von allen befragten Nutzern als positiv bewertet (bezogen auf die Tools EXMARaLDA und PRAAT). Die Konfigurationsmöglichkeiten in EXMARaLDA wurden von den Nutzern, die mit diesem Tool arbeiten, als gut bewertet, allerdings wurde für die Exploration der Daten ein höherer Interaktivitätsgrad gewünscht, in dem Sinne, dass die Markierung einzelner Annotationswerte direkt aus der Partitur heraus möglich sein sollte, nicht erst über den Umweg eines Konfigurationsmenüs.
- 2) Die Darstellung von ANNIS wird als nicht ausreichend und sehr unübersichtlich bewertet.

- 3) Kombinationen von Darstellungen sind erwünscht und vielfältig. Es wurden als wichtig bis sehr wichtig genannt: Partitur + Baum, Partitur + Koreferenz, Partitur + RST, Koreferenz + RST, Partitur + Tabelle²⁵; (außerdem eine vollfunktionale Integration von RST + PRAAT). Bei der Frage nach den gewünschten Visualisierungsformen/-kombinationen ist von Visualisierungen ausgegangen worden, die den Nutzern u.a. aus Annotationstools bekannt sind (Partitur: EXMARaLDA, PRAAT und Toolbox; Baum: TIGER; Koreferenz MMAX; RST (RSTTool)). Darüber hinausgehende Visualisierungen sind nicht ausdrücklich gewünscht worden; nachdem die Nutzer jedoch mit Skizzen konfrontiert wurden, die eine Kookkurenz von Daten schematisch darstellten, wurde diese Darstellungsform ebenfalls als sehr sinnvoll angesehen.
- 4) Für die Exploration des PCC in allen Ebenen wäre eine kombinierte Darstellug von Informationsstruktur, Syntax, RST und Koreferenz vorteilhaft.
- 5) Es werden zeitbasierte und elementbasierte Darstellungen gewünscht, auch innerhalb eines Projektes, mit der Möglichkeit, zwischen den Darstellungen hin- und herzuschalten. Dabei wurde jedoch angemerkt, dass bei der Exploration mittels einer zeitbasier-ten Darstellung teilweise auch sehr spezifische Abfragen und spezielle Einstellungen möglich sein müssten, ähnlich wie in PRAAT.
- 6) Minimale Bezugseinheiten für eine Annotation sind bei der zeitbasierten Darstellung die Millisekunde, bei der elementbasierten Darstellung das Phonem; weitere element-basierte Bezugseinheiten sind z.B. Morphem, Silbe, Token. Morphem und Silbe wer-den jedoch bei der Annotation nicht gesondert in eine Spalte geschrieben, sondern tokenbezogen realisiert.
- 7) Es wird die Anforderung formuliert, während der Exploration auch auf die nicht-textbasierten Rohdaten (Audiodateien, Videodateien, Bild) zugreifen zu können.
- 8) Es wird als wichtig betrachtet, während der Exploration auf die Metadaten zu einem Dokument zugreifen zu können, sofern sie erhoben wurden. Metadaten zu Anno-tationen (z.B. wer hat die Primärdaten auf welchem Annotationsset wann annotiert) sind nur in einem Projekt erhoben worden und werden für die Exploration nicht benötigt.
- 9) Die Textlänge der Dokumente variiert innerhalb der meisten Projekte wenig

²⁵ Unter einer Tabelle wurde ganz allgemein eine Layoutform verstanden, die eine Anordnung zweier Dokumente/Einheiten in nebeneinander liegenden Spalten ermöglicht. Diese Darstellungsform wurde speziell in Projekten, in denen verschiedene Sprachen miteinander verglichen werden sollen, als positiv bewertet, ohne dass jedoch eine Spezifizierung der Form erfolgft wäre.

(1 Ausnahme: 2 bis 150 Token), zwischen den Projekten dagegen stärker (zwischen 2 Token und einer geschätzten DIN A4 Seite).

- 10) Die Mehrzahl der Nutzer schätzt es als äußerst wichtig ein, eigene Einstellungen vornehmen zu können. Dies wurde in Gesprächen mehrmals betont.
- 11) Für die voraussichtliche Anzahl an annotierten Dokumenten gaben die Nutzer Folgendes an: mehr als 1000 Dokumente (B1), 400 Dokumente (B2), bis zu 10 000 Dokumente (D2).

Im Rahmen dieser Arbeit kann selbstverständlich nicht auf alle (potenziellen) Nutzer im Detail eingegangen werden. Es wird im Folgenden eine Zielgruppe definiert:

Allgemein ist anzumerken, dass die reinen Anwender (und nicht die Administratoren) fokussiert werden, hier speziell Linguisten, die reich annotierte linguistische Daten explorieren möchten. Das können sowohl SFB-Projektbeteiligte sein, aber auch beliebige andere Nutzer, die sich für die Daten interessieren. Es werden dabei häufige Nutzer und gelegentliche Nutzer erwartet sowie Computerexperten und Normalanwender.

Schwerpunktmäßig soll der motivierte interneterfahrene Normalanwender betrachtet werden.

2 Zwei linguistische Fragestellungen (Szenarien)

1 Szenario 1: Informationsstruktur und Verbstellung im Althochdeutschen

Das Projekt B4 untersucht die „Interaktion informationsstruktureller und grammatischer Bedingungen bei der Herausbildung von Wortstellungsregularitäten in den germanischen Sprachen“(aus der Projektbeschreibung)²⁶.

Dabei geht es unter anderem darum, herauszufinden, wie es zu der heutigen Vielfalt an Wortstellungsmustern (Reihenfolge von Objekt und finitem Verb) in den germanischen Sprachen gekommen ist, obwohl diese sich aus einer gemeinsamen Wurzel heraus entwickelt haben. Während das Deutsche gekennzeichnet ist durch eine Verbzweitstellung (OV), ist z.B. im Englischen die Reihenfolge Verb-Objekt (VO) obligatorisch. Es wird der Einfluss von informationsstrukturellen Faktoren diskutiert, da in vielen Sprachen die Fokussierung einer Konstituente mit einer Änderung der Wortstellung verknüpft ist. Dies für die germanischen Sprachen zu überprüfen ist ein Ziel des Forschungsprojekts (vgl. ebd.). Im Folgenden wird eine konkrete (vereinfachte) Fragestellung definiert, die mit Hilfe einer

²⁶http://www.sfb632.uni-potsdam.de/projects.php?lang=de&project_name=B4 (Projektbeschreibung)

Datenbankabfrage und Sichtung der Daten in ANNIS beantwortet werden soll.

Eine Arbeitshypothese lautet, dass die Tatsache, dass die Stellung des finiten Verbs im frühesten Germanischen so stark variiert, auf die Funktion zurückzuführen ist, die das finite Verb in der informationsstrukturellen Gliederung der Aussage einnimmt. Um diese Hypothese zu überprüfen, soll in den erhobenen Daten geprüft werden, (a) ob es bestimmte Verbstellungsmuster (Verberststellung, Verbzweitstellung, Verbendstellung) in Abhängigkeit vom pragmatischen Status von Diskursreferenten (Ersteinführung, Weiterführung) gibt und (b) ob eine Variante häufiger vorkommt als die andere.

Es können folgende Schritte unternommen werden, um die o.g. Fragen zu beantworten:

zu (a): *Welche Verbstellungsmuster kommen bei Ersteinführung von Diskursreferenten überhaupt vor?*

1. Formuliere in der Abfragesprache von ANNIS eine Suche nach allen Dokumenten mit einem Diskursreferenten in Ersteinführungs-Status: „givenness=new“.
2. Zeige die Ergebnisse in einer Liste an: Klicke auf „Suche“.
3. Identifiziere in der Ergebnisliste alle Annotationen mit dem Wert „givenness=new“.
4. Identifiziere in der Ergebnisliste alle Annotationen mit dem Wert „vfin“ (das finite Verb).
5. Identifiziere in der Ergebnisliste die Position aller „vfin“.
6. Vergleiche die Ergebnisse der Liste hinsichtlich der Positionen von „vfin“ und von „new“.

zu (b): *Kommt eine Variante häufiger vor als die andere?*

1. Bestimme die Anzahl der Vertreter jedes Musters.
2. Vergleiche die Anzahl der gefundenen Items.

Die folgenden Textbeispiele stellen eine (hypothetische) Ergebnismenge für eine Suchanfrage nach finitem Verb (gf=vfin) und neuem Diskursreferenten (givenness=new) dar:

<i>T_035__29 (733501885)</i>	uuarun thô hirta In thero landskeffi .'
<i>T_036__11-12 (149710514)</i>	thô sliumo uuard thar mit themo engile / menigi himilisches heres .'
<i>T_037__23 (784152424)</i>	senonu tho uuas man In hierusalem .'
<i>T_038__22(-2062701212)</i>	uuas thô thâr anna uuizzaga
<i>T_039__17(502633760)</i>	thô thaz gihorta herodes ther cuning
<i>T_051__16-17(1523747531)</i>	thero uuas andreas bruoder / simonis p&ri

2 Szenario 2: Modelle der Textstruktur

„[...] having a picture of syntax, co-reference, and sentence-internal information structure at one's disposal should aid in finding models of discourse structure [...].“ (Stede 2004)

Die Erstellung des Potsdamer Commentary Corpus ist zum einen motiviert durch die Frage nach diskurskonstituierenden Faktoren und zum anderen durch die Suche nach einem diskursbasierten Modell der Informationsstruktur (ebd.). Die Explorationsstrategien, die bei der Suche in den Daten angewendet werden, sind dementsprechend komplex. Es soll hier kein konkretes Szenario erstellt werden, sondern vielmehr festgehalten werden, dass während der Exploration eines Datensatzes so viele linguistische Annotationen zueinander in Beziehung gesetzt werden sollen wie möglich. Dabei spielt auch die Position von Einheiten eine Rolle. Während sich in Szenario 1 die Exploration eher auf der Ebene der Ergebnisliste abspielt, sind hier die Detailansichten von größerer Bedeutung.

Im Folgenden einer der zehn annotierten Texte im PCC, die hinsichtlich der Ebenen Syntax (Tagset pcc11-tiger), Koreferenz (Tagset pcc11-coref), Rhetorical Structure (Tagset pcc11-rst) und Informationsstruktur (Tagset pcc11-is):

„Glückliche Hühner Um die deutschen Legehennen ist heftiger politischer Streit entbrannt . Bundesagrarministerin Renate Künast will das Halten der Tiere in engen Legebatterien bereits vom Jahr 2006 an verbieten . In den EU-Nachbarländern soll das erst fünf Jahre später gelten . Eier-Produzenten aus der ganzen Republik machen gegen Künasts Pläne mobil . Die Betriebe im Osten fürchten , dass die hohen Investitionen , die sie in moderne Legebatterien nach europäischem Standard gesteckt haben , umsonst gewesen sind . Im Westen herrscht die Sorge vor , ausländische Konkurrenten könnten dann mit Billig-Eiern aus Legebatterien den deutschen Markt überschwemmen . Beide Bedenken sind nicht einfach von der Hand zu weisen . Ministerin Künast muss zumindest über längere Übergangsfristen nachdenken . Am Grundsatz , die Käfighaltung abzuschaffen , sollte sie nicht rütteln . Artgerechte Tierhaltung gehört schließlich zu den Eckpunkten der von ihr propagierten Agrarwende . Und schließlich greifen immer mehr Verbraucher beim Einkauf bewusst zu den tierfreundlich erzeugten Eiern .“

Quelle: ANNIS: urml.maz-5715.anno (877600497); Stand: 4.11.2006.

3 Gegenstandseingrenzung

Die verschiedenen Annotationen und entsprechenden Rohdaten in vielfältiger Weise visuell in Beziehung setzen zu können, ist die Hauptanforderung der Nutzer.

Metadaten spielen bisher im Rahmen des SFB 632 so gut wie keine Rolle. Nur wenige Metadaten sind erhoben worden. Es ist anzumerken, dass die Unterscheidung zwischen Metadaten und Annotationen nicht getroffen wurde, mit der Folge, dass Informationen, die sich auf das gesamte Dokument beziehen und damit laut meiner Definition zu den Metadaten gehören, sich innerhalb der Annotationen wiederfinden und damit auch nur als Annotation visualisiert werden können (Bsp. Projekt B4). Die spezifische Visualisierung von Metadaten ist damit erst einmal sekundär. Allerdings sollte natürlich ein Zugriff auf die Metadaten eines Dokumentes an geeigneter Stelle im Explorationsprozess ermöglicht werden. (Aufgrund der Zuordnung zum Dokument bietet sich eine Zugriffsmöglichkeit innerhalb der Detailansicht an, wäre aber auch in einer Ergebnisliste denkbar, ebenfalls dem Dokument zugeordnet). Es wird ein Schwerpunkt gesetzt auf die Art und Weise, in der mehrfach annotierte heterogene Daten visualisiert werden können.

Im weiteren Verlauf werde ich mich auf die Visualisierung von Primärdaten und Annotationen beschränken.

Für die Visualisierung von Primärdaten und Annotationen wurden folgende Darstellungen „gewünscht“: Interaktive Partitur, Partitur + Baum, Partitur + Koreferenz, Partitur + RST, Koreferenz + RST, Tabelle + Partitur. Ich wähle die gewünschte PCC-Visualisierung (Informationsstruktur (Partitur, intralinear) + Syntax + RST + Koreferenz)) als *worst case*. Die zu entwickelnde Visualisierung soll in der Lage sein, die vier Komponenten (interaktive Partitur, Baum, Koreferenz und RST) abzubilden und miteinander in Beziehung zu setzen. Alle anderen Darstellungskombinationen (außer der unspezifizierten Tabelle) werden mit dieser Visualisierung abbildbar sein. Nicht benötigte Komponenten sollten dabei vom Nutzer weggeblendet werden können.

Audio, Video- und Bilddaten schließe ich aus meiner Betrachtung aus. Diese Daten können mit einem einfachen Link zugänglich gemacht werden und in einem kleinen Abspiel-/Anzeigebereich visualisiert werden. Eine komplexe Synchronisierung von Audiodaten und Annotationen den Rahmen dieser Arbeit sprengen würde.

Evaluation/Bewertung ausgewählter Visualisierungswerkzeuge

1 Auswahlkriterien und Bewertungsrahmen

Auswahlkriterien

Web-Interfaces bieten für eine Visualisierung von Daten selten mehr als (einfache und komplexere) Ergebnislisten (z.B. COSMAS II, DWDS) . Detailansichten mit einer Darstellung von Annotationen kommen darin nur bedingt vor. Sie sind als Ausgangspunkt für die Visualisierung reich annotierter Daten also nur insofern geeignet, als sie einen wichtigen Schritt im Information Retrieval Prozess abdecken: die Auflistung von Suchergebnissen auf eine Suchanfrage.

Für Detailansichten werden jedoch ausgereifere Visualisierungen benötigt, die in der Lage sind, eine größere Anzahl von Annotationsebenen darzustellen. Solche Visualisierungsformen werden bereits in *lokal* zu installierenden Annotationstools genutzt. Diese Visualisierungen sind nicht primär für die *Exploration* der Daten konzipiert worden. Es scheint jedoch aus folgendem Grund legitim, diese Visualisierungen im Kontext der Exploration zu betrachten: Während des Annotationsprozesses ist es wichtig, sich zu vergewissern, welche Daten bereits annotiert sind und ob sie richtig annotiert sind. Auch im Annotationsprozess findet also eine Art Suche nach den annotierten Daten statt. Der Fokus liegt hierbei auf der korrekten Zuweisung einer Annotation zum Originaltext und nicht wie bei der Exploration auch auf der Erforschung komplexerer Zusammenhänge zwischen den Annotationen selbst. In beiden Fällen muss jedoch in mehr oder weniger schneller Abfolge auf Annotationen verschiedener Ebenen zugegriffen werden können. Kurz, ich betrachte die Annotationstools als geeigneten Ausgangspunkt für die Visualisierung mehrfach annotierter Daten, speziell für die Phase der Darstelllung im Information Retrieval Prozess.

Bewertungsrahmen für die Bewertung von Visualisierungen ausgewählter Tools

Es ist kein vergleichender Test zur Visualisierung reich annotierter linguistischer Daten möglich, da die betrachteten Tools jeweils nur Visualisierungen bereitstellen, die für die unterstützten Datentypen erforderlich sind, und keines der betrachteten Tools alle im SFB 632/PCC erforderlichen Datentypen unterstützt. Die Tools werden hinsichtlich der folgenden Aspekte bewertet: (a) Anzahl an darstellbaren Annotationsebenen (b) Vergleich verschiedener Annotationsebenen, (c) Die Heterogenität der Daten, (d) die Darstellung längerer Texte, (e) die Druckbarkeit und (f) die Visualisierung von Position.

2 ANNIS

Kurzbeschreibung

Das Tool ist bereits in Kapitel 2 kurz vorgestellt worden. Es soll hier noch einmal etwas genauer betrachtet und hinsichtlich der Anforderungen bewertet werden. Schwierigkeiten bei der Nutzung sollen Aufschluss geben über zu vermeidende Fehler bei der Visualisierung der Daten.

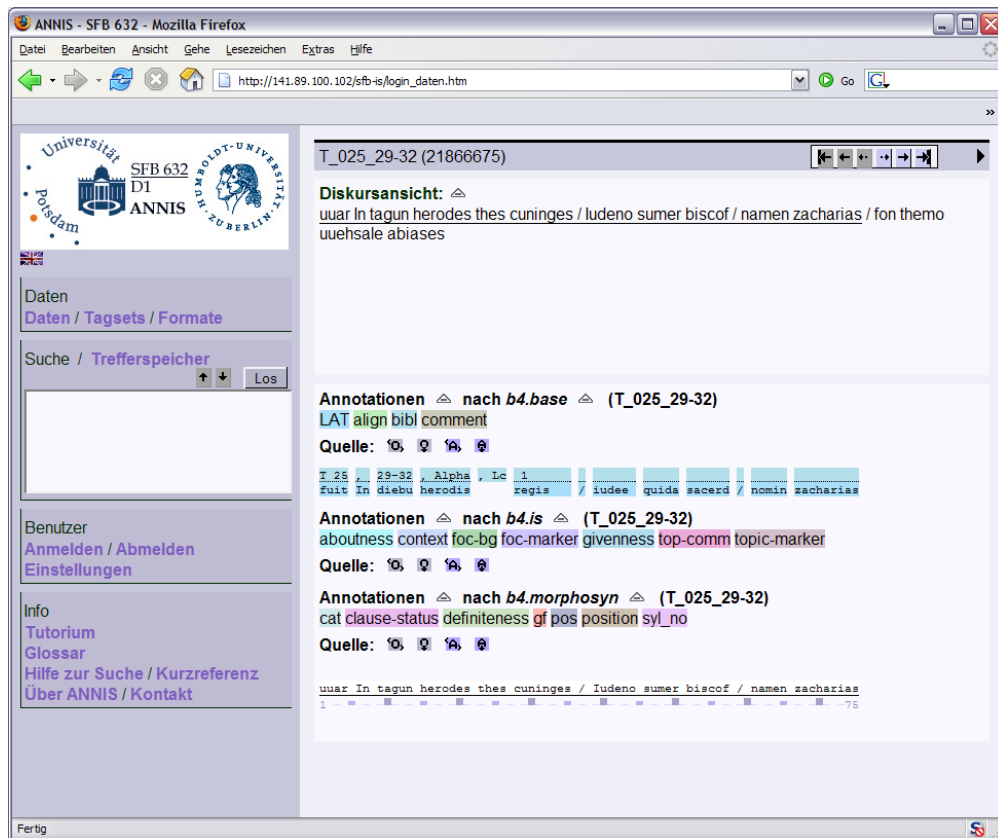


Abbildung 2: Screenshot ANNIS (Detailansicht eines Dokuments)

Umfang und Art der dargestellten Daten: Es werden textuelle Rohdaten und Annotationen visualisiert. Folgende Datentypen können dargestellt werden: Informationsstruktur (und andere beliebige Attribut-Wert-Paare), RST, Syntax und Koreferenz.

Datenvisualisierung: Die verschiedenen Daten werden in ANNIS wie folgt dargestellt:

- textbasierte Rohdaten bzw. das orthografische Transkript werden als Fließtext dargestellt
- sämtliche Annotationen (+ der zugehörige Textabschnitt) werden in einer Ebenendarstellung visualisiert.

Die Ebenenvisualisierungen für RST, Koreferenz und Tiger stellen besondere Problembe-
reiche dar. Die folgenden Screenshots zeigen jeweils ein Beispiel für die Darstellung dieser
Annotationsebenen.

elaboration						
	satellite					
	evaluation					
	nucleus					
	volitional-cause					
	satellite					
	preparation					

nucleus		sa	tel	lite						
---------	--	----	-----	------	--	--	--	--	--	--

Annotationen nach *pcc11-tiger* (tiger.ID_maz-5715.anno)

Glückliche Hühner Um die deutschen Legehennen ist heftiger politischer Stre

Abbildung 4: ANNIS (Syntax) Beispiel aus Dipper et al. (2004)

Die hierarchische Struktur in Abbildung 3 ist nur schwer erkennbar – zumal eine (vom Nutzer zwar definierbare, aber immer noch fixe) Breite den Textausschnitt losgelöst von seinem hierarchischen Kontext zeigt und dieser Kontext erst über ein erneutes Laden der

Seite hergestellt werden kann. Etwas besser lässt sich eine Hierarchie des *Syntax-Bereiches* interpretieren – wenn auch bei weitem nicht so gut wie der entsprechende Graph.

Abbildung 4 (Beispiel entnommen aus Dipper et al. 2004:270), kann den Unterschied demonstrieren. Für die Darstellung von Koreferenz ist die bisherige Ebenenvisualisierung nicht geeignet (s. Abbildung 5).

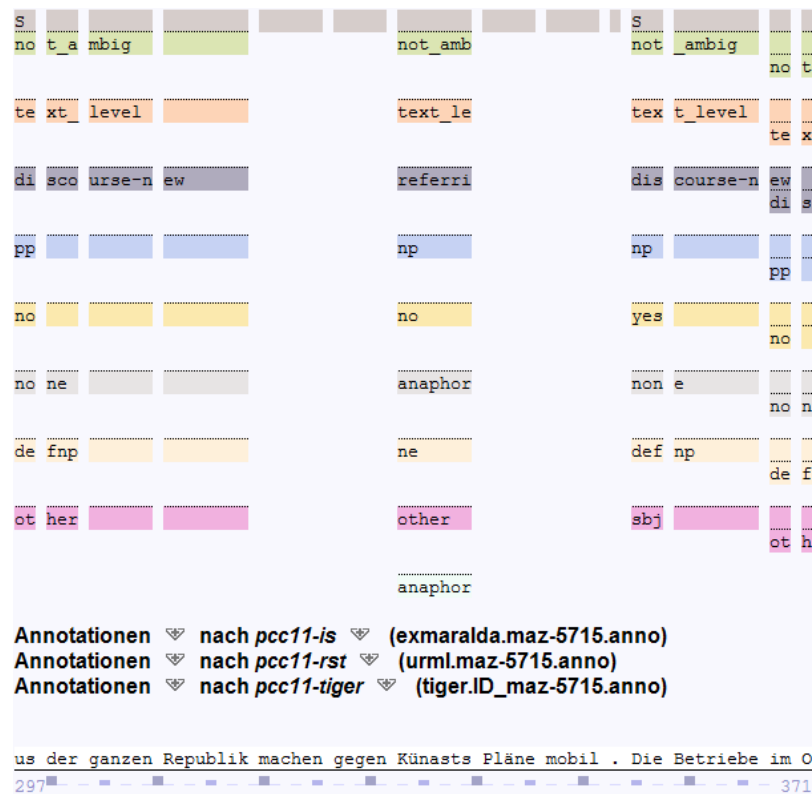


Abbildung 5: ANNIS (Koreferenz)

Interaktion: Neben einer textbasierten Suche gibt es diverse Interaktionsmöglichkeiten vom Typ der direkten Manipulation. Folgende Interaktionen sind innerhalb einer Detailansicht möglich:

- Einzelne Annotationssets können ein- und ausgeblendet werden.
- Es lassen sich Segmente mit einem Mouseover über einen Annotationswert einsehen (dies ist allerdings nicht immer praktikabel (s.u.).
- Es lassen sich per Mouseover über einem (ausgelagerten) Zeilenlabel die Textsegmente hervorheben, die hinsichtlich der Annotationsebene annotiert wurden.
- Über die Diskursansicht kann an eine bestimmte Stelle innerhalb der Partitur gesprungen werden (allerdings kann in der Ebenendarstellung selbst nicht weitergescrollt werden).
- Es kann der Original-Code angezeigt (und heruntergeladen) werden.

Konfigurationsmöglichkeiten: Der Nutzer erhält die Möglichkeit, diverse Einstellungen vorzunehmen, um das Nutzerinterface an seine Bedürfnisse anpassen zu können. Dazu gehören z.B. die Breite der Partitur und die Festlegung der Nutzeraktion (mouseover, press, click)

Bewertung

In den geführten Interviews und bei der Gelegenheit, einem ungeübten linguistischen Nutzer bei der Nutzung von ANNIS über die Schulter zu sehen (*thinking loud*), hat sich Folgendes gezeigt:

- (a) die subjektive Zufriedenheit mit dem Tool ist eher gering;
- (b) die Nutzeroberfläche stellt den Nutzer vor diverse Fragen, die er sich nicht unmittelbar erschließen kann, und die ihn vom eigentlichen Ziel, nämlich die Daten zu sichten, abhalten. Diese Fragen manifestierten sich in z.T. längeren Pausen bei der Interaktion (gefolgt von: „ach so, vielleicht hier“ - „nein, doch nicht“)²⁷. Während davon auszugehen ist, dass die Lokalisierung bestimmter Funktionsauslöser (speziell die Buttons für das Ein-/Ausblenden der Annotationen bzw. Tagsets) irgendwann memoriert wird, sind andere Darstellungsprobleme von Dauer. Diese beziehen sich im Wesentlichen auf die Perzeption der Daten in der Ebenendarstellung. Die Probleme werden im Folgenden aufgeführt:
 - (1) *Farben*: Die Vielzahl der Farben wirkt irritierend; die Zahl ist so groß, dass einzelne Zuordnungen von Farbe und Annotationsebene nicht memoriert werden. Farbkodierung: die Farben sind nicht distinktiv, sie unterscheiden sich teilweise so wenig, dass eine eindeutige Zuweisung von Annotationsset zu Ebene nicht möglich ist (getestet für TFT-Bildschirm) – die Zuweisung kann auch nicht etwa in einem zweiten Schritt über einen Positionsvergleich erfolgen, da die Abfolge der Annotationssets eine andere ist als die der Zeilen. Als Ersatz kann per Mouseover über einem Annotationswert die zugehörige Annotationsebene eingeblendet werden. Dies wird allerdings nicht als intuitiv empfunden.
 - (2) *Fehlender Bezug zur Referenzebene*: Der Abstand zwischen dem Bezugstext und den Annotationen ist teilweise zu groß; bei Einblenden aller Annotationssets (es können immer nur einzelne Sets ein- und ausgeblendet werden, nie einzelne Ebenen eines Sets) kann es sein, dass die Referenzzeile gänzlich vom Screen verschwindet und nicht mehr zu den Annotationen in Beziehung zu setzen ist, womit eine Interpretation der Annotationen nicht mehr möglich ist.

²⁷ im Übrigen an denselben Stellen, die auch mir bei der Erst-/Zweit- und Drittnutzung des Tools Fragen aufgaben.

- (3) Der *Vergleich zwischen zwei Werten*, die sich auf Ebenen von verschiedenen Annotationssets befinden, ist aufgrund ihrer räumlichen Trennung erschwert.
- (4) *Visuelle „Löcher“*: Der Abstand zwischen den einzelnen Zeilen eines Annotationssets²⁸ ist teilweise zu groß; es entstehen Löcher, die eine Orientierung innerhalb der Seite stark erschweren, da durch den erzeugten Abstand nachstehende oder vorangestellte Annotationssets näher erscheinen als die Inhalte des eigenen Annotationssets. Die visuelle Clusterung widerspricht hier also der informationellen Clusterung, das heißt: die Gruppen, die wahrgenommen werden, entsprechen nicht den intendierten Gruppen.
- (5) Die *vertikale Segmentierung* entspricht nicht der informationellen Segmentierung; Segmente, die sich über mehr als ein Token erstrecken, werden nicht direkt abgebildet, sondern erst sichtbar nach onmouseover, und zwar auf der (je nach Position des Annotationssets mehr oder weniger) räumlich entfernten Ebene des Bezugstextes.
- (6) Lesbarkeit: die Trennung der Wörter, die durch die visuelle Segmentierung bedingt ist, vermindert die Lesbarkeit der Annotationen gravierend.
- (7) Die Navigation in der Ebenendarstellung erfolgt nur über die Diskursansicht, bzw. über „Zurück“- und „Weiter“-Buttons in der Titelleiste, nicht über die Partitur selbst. Schnell zum nächsten Token bzw. Segment zu gehen, ist nicht möglich, da die Ladezeit bei längeren Texten sehr lang sein kann.

Hinsichtlich der formulierten Anforderungen lässt sich festhalten:

Die *Anzahl an darstellbaren Annotationsebenen* ist recht groß, allerdings geht der (unerlässliche) Bezug zur Referenzebene verloren, wenn zu viele Ebenen eingeblendet sind. Teilweise kann dieses Problem durch das Ausblenden von Annotationsebenen umgangen werden.

Ein *Vergleich verschiedener Annotationsebenen* wird dadurch erschwert, dass sich einzelne Zeilen eines Annotationssets nicht verschieben lassen. Ein Vergleich konkurrierender Annotationen innerhalb eines Annotationssets ist indes möglich.

Die *Heterogenität* der Daten kommt in den derzeitigen Visualisierungen nicht zum Ausdruck: Die spezifische Visualisierung hierarchischer Daten (speziell RST) wird nicht unterstützt. Eine spezielle Visualisierung von Koreferenzdaten ist nicht möglich.

Praktikabel für die Darstellung *längerer Texte* ist die gleichzeitige Ansicht von Fließtext (Diskursansicht) und Ebenendarstellung, inklusive der Navigation über den Fließtext (die

28 Dieser Abstand kann allerdings nicht reduziert werden, da er in der Anzahl der Annotationsebenen gründet, die an irgendeiner Stelle innerhalb der Partitur gefüllt sind und folglich existieren müssen, damit es nicht zu einem Verrutschen der Ebenen kommt.

allerdings durch zusätzliche Navigationsoptionen erweitert werden müsste).

Druckbarkeit: Die visualisierten Annotationen lassen sich nur unter Informationsverlust ausdrucken, da der Segmentbezug einer Annotation nur temporär per Mouseover dargestellt wird und somit nicht mit gedruckt werden kann.

Visualisierung von *Position*: Eine Segmentnummerierung ist angelegt. Die Nummerierung erfolgt auf Buchstabenebene; diese ist allerdings für den Nutzer nicht primär relevant (sofern die visuelle Segmentierung sich an den Token orientiert und nicht an den Buchstaben). Eine Änderung des „Maßstabs“ auf Tokensegmente ist nicht möglich.

Referenzen

Dipper (2004)

3 EXMARALDA

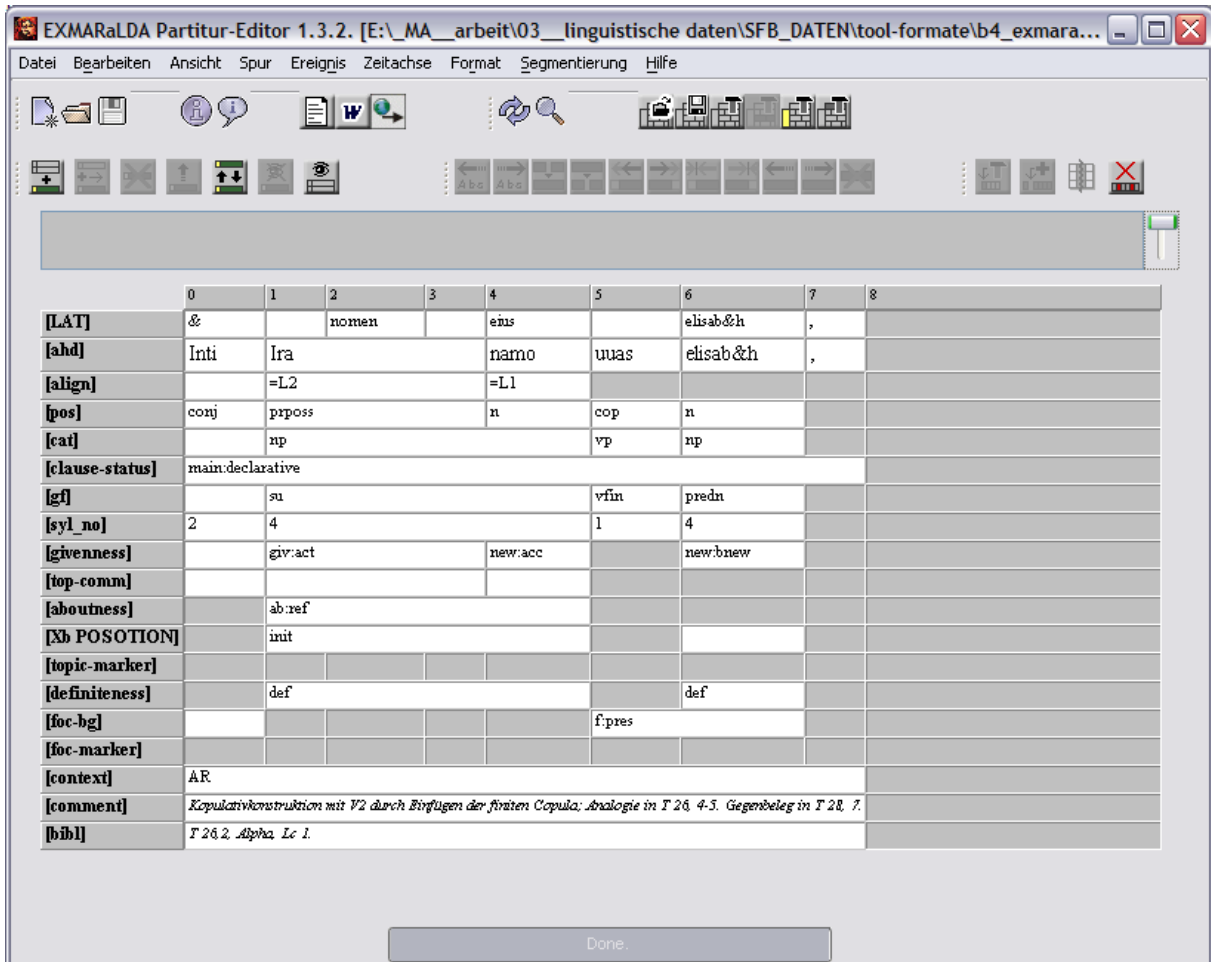
Kurzbeschreibung

EXMARaLDA ("Extensible Markup Language for Discourse Annotation") ist ein frei verfügbares, lokal zu installierendes Softwaresystem²⁹, das von Thomas Schmidt im Rahmen des SFB 538 "Mehrsprachigkeit" an der Universität Hamburg entwickelt wurde. Das System besteht aus den drei (getrennt zu installierenden) Komponenten Partitur-Editor, Corpus-Manager und einem Suchwerkzeug (SQUIRREL; eine neuere Version ZECKE befindet sich in der Testphase) und wird seit 2001 kontinuierlich weiter entwickelt. Der EXMARaLDA Partitur-Editor wird von mehreren Projekten des SFB 632 zum Annotieren von Rohdaten genutzt.

Umfang und Art der dargestellten Daten: Sowohl Text als auch Audio und Videodaten können dargestellt werden. Der Primärtext kann auf beliebig vielen Ebenen mit beliebig vielen Attribut-Wert-Paaren beschrieben werden.

Datenvisualisierung: Die Darstellung der Daten beruht auf einer (element-basierten) Partiturnotation. In dieser Form der Visualisierung, die strukturell gesehen nichts anderes ist als eine Tabelle, können Primärtexte in einer Zeile eingefügt werden und diese auf den darunter liegenden Zeilen beschrieben werden. Die vertikale Segmentierung können dabei vom Nutzer frei gewählt werden.

²⁹ downloadbar unter <http://www1.uni-hamburg.de/exmaralda/>



EXMARaLDA Partitur-Editor 1.3.2. [E:_MA_arbeit\03_linguistische daten\SFB_DATEN\tool-formate\b4_exmara...]

Datei Bearbeiten Ansicht Spur Ereignis Zeitachse Format Segmentierung Hilfe

	0	1	2	3	4	5	6	7	8
[LAT]	&		nomen		eins		elisab&h	,	
[ahd]	Inti	Ira			namo	uuas	elisab&h	,	
[align]		=L2			=L1				
[pos]	conj	prposs			n	cop	n		
[cat]		np				vp	np		
[clause-status]	main:declarative								
[gf]		su				vfin	predn		
[syl_no]	2	4				1	4		
[givenness]		giv:act			new:acc		new:bnew		
[top-comm]									
[aboutness]		ab.ref							
[Xb POSOTION]		init							
[topic-marker]									
[definiteness]		def					def		
[foc-bg]						f:pres			
[foc-marker]									
[context]	AR								
[comment]	Kopulativkonstruktion mit V2 durch Einfügen der finiten Copula; Analogie in T 26, 4-5. Gegenbeleg in T 28, 7.								
[bibl]	T 26.2. Alpha, Lc 1.								

Done

Abbildung 6: Screenshot aus EXMARaLDA 1.3.2.

Interaktionsgrad: Es sind diverse Interaktionen des Typs Direkte Manipulation implementiert. Es können Zeilen verschoben ein-/ausgeblendet und hervorgehoben werden. Es können einzelne Zellen und Spalten markiert und die Annotationen direkt in ihrer Position bearbeitet werden. Spalten können geteilt und in ihrer Breite verändert werden. Der Interaktionsgrad ist insgesamt sehr hoch. Die Aktionen sind komfortabel über standardkonforme Kontextmenüs auszulösen, die über die rechte Maustaste eingeblendet werden.

Konfigurationsmöglichkeiten: Die Einstellungsmöglichkeiten sind sehr umfangreich. Sämtliche Funktionen, die direkt am Objekt interaktiv ausgeführt werden können, sind auch über die normale Menüleiste erreichbar. Darüber hinaus können noch etliche weitere Einstellungen vorgenommen werden, die hier nicht einzeln aufgelistet werden sollen.

Bewertung

Anzahl an darstellbaren Annotationsebenen: Die Anzahl der gleichzeitig darstellbaren Annotationsebenen ist nicht nach oben beschränkt.

Vergleich verschiedener Annotationsebenen: Ein Vergleich verschiedener Annotations-ebenen ist aufgrund der Gleichzeitigkeit der Darstellung möglich.

Heterogenität: Es können zwar beliebige Attribut-Wert-Paare in der Partitur-Visualisierung dargestellt werden, jedoch ist die Visualisierung von Bäumen als Ebenen nicht optimal; Koreferenz kann in der Partitur-Darstellung ebenfalls nicht optimal visualisiert werden, da bei längeren Texten die Gleichzeitigkeit der Darstellung bezogen auf die Spalten nicht mehr gegeben ist.

Darstellung längerer Texte: Längere Texte können zwar visualisiert werden, jedoch ist immer nur ein Ausschnitt sichtbar, der sich durch vertikales Scrollen verschieben lässt. Für die Visualisierung von Werten, die innerhalb eines Textes weit auseinander liegen, ist die Partiturdarstellung in EXMARaLDA nicht geeignet.

Visualisierung von Position: Die Position wird über eine Segmentnummern ermöglicht, die in der Titelzeile fixiert sind und damit auch dann noch im direkten Blickfeld liegen, wenn bedingt durch eine große Anzahl an Zeilen die Partitur in der vertikalen gescrollt werden muss.

Referenzen:

<http://www1.uni-hamburg.de/exmaralda/>

4 MMAX 2

Kurzbeschreibung

Das Tool MMAX ist von Christoph Müller speziell für die Annotation von anaphorischen und Bridging-Beziehungen entwickelt worden (Müller & Strube 2001).

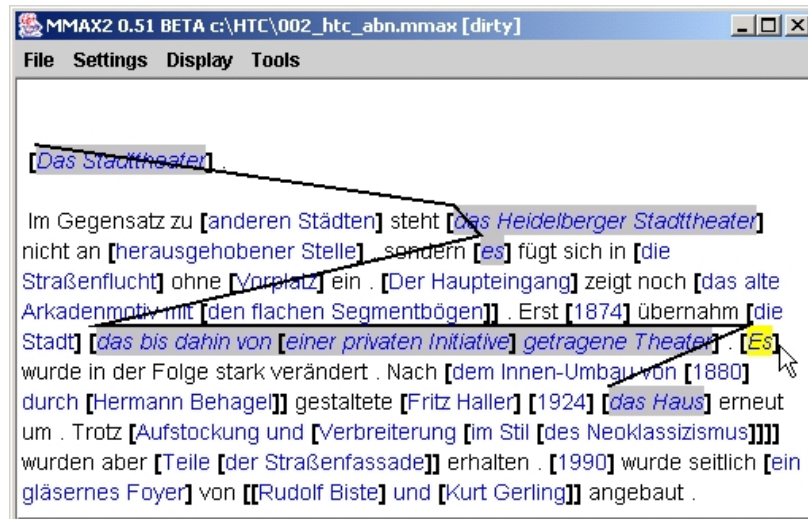


Abbildung 7: Screenshot aus MMAX 2.0 (coreference)



Abbildung 8: Screenshot aus MMAX 2.0 (Bridging)

Umfang und Art der dargestellten Daten: Es können Textdaten mit Annotationen versehen werden. Dabei werden so genannte „markables“, das sind auswählbare und beschreibbare Textsegmente, mit einem oder mehreren Attributen versehen. Weiterhin können Relationen zwischen Textsegmenten annotiert werden – und zwar sowohl anaphorische als auch Bridgingrelationen.

Datenvisualisierung: Die Annotation der Relationen wird unterstützt durch folgende Visualisierungen: Der normale Text wird in schwarzer Schrift dargestellt; Markables der Koreferenz-Ebene werden in blauer Schrift und – wenn sie sich in einer Koreferenz-Beziehung befinden – zusätzlich kursiv dargestellt. Eingefügt in den Fließtext sind so genannte

„markable handles“ (schwarze eckige Klammern), die als maussensitive Elemente die Auswahl eines Markables ermöglichen. Ein ausgewähltes Markable wird durch gelbe Hintergrundfarbe hervorgehoben. Relationen werden durch Verbindungslinien zwischen den verbundenen Markables visualisiert, deren Erscheinungsbild sich konfigurieren lässt (z.B. gerade Linie, gewölbte Linie, Linienstärke).

Interaktion: MMAX bietet mehrere Interaktionsmöglichkeiten des Typs direkte Manipulation sowie Formulare bzw. Menüauswahl zur Zuweisung von Attributen; Aktionen können nicht nur über die linke, sondern auch über die rechte Maustaste (+Kontextmenü) initiiert werden. Der Interaktionsgrad ist also recht hoch.

Konfigurationsmöglichkeiten: Die Visualisierung kann hinsichtlich der Variablen Schriftfarbe, Schriftfont, Schriftstil, Hintergrundfarbe, Linienstärke/-typ/-farbe vom Nutzer verändert werden. Sogar die Klammerung um die einzelnen Markables ist konfigurierbar. Eigene XSLT-Stylesheets können eingebunden werden.

Bewertung

Anzahl an darstellbaren Annotationsebenen: Es können beliebig viele sogenannte „Markable Levels“ erzeugt werden, mit eigener Segmentierung und Formatierung; eine gleichzeitige Darstellungen von verschiedenen Annotationsebenen im Zusammenhang mit dem Text ist jedoch nicht möglich.

Vergleich verschiedener Annotationsebenen: Ein direkter visueller Vergleich von Annotationsebenen ist nicht möglich.

Heterogenität der Daten: Es können Attribut-Wert-Paare und Relationen visualisiert werden, hierarchische Strukturen können nur über verschachtelte Klammerung dargestellt werden.

Darstellung längerer Texte: Die Fließtextdarstellung ist für längere Texte gut geeignet.

Visualisierung von Position: Die Visualisierung von Position ist nicht möglich.

Referenzen

Müller (2006)

Müller (2005)

5 RSTTool

Kurzbeschreibung

Das RSTTool von Mick O'Donnell wurde mit dem Ziel entwickelt, die Annotation von Texten hinsichtlich der Rhetorical Structure nach Mann & Thompson (1987)³⁰ grafisch zu unterstützen.

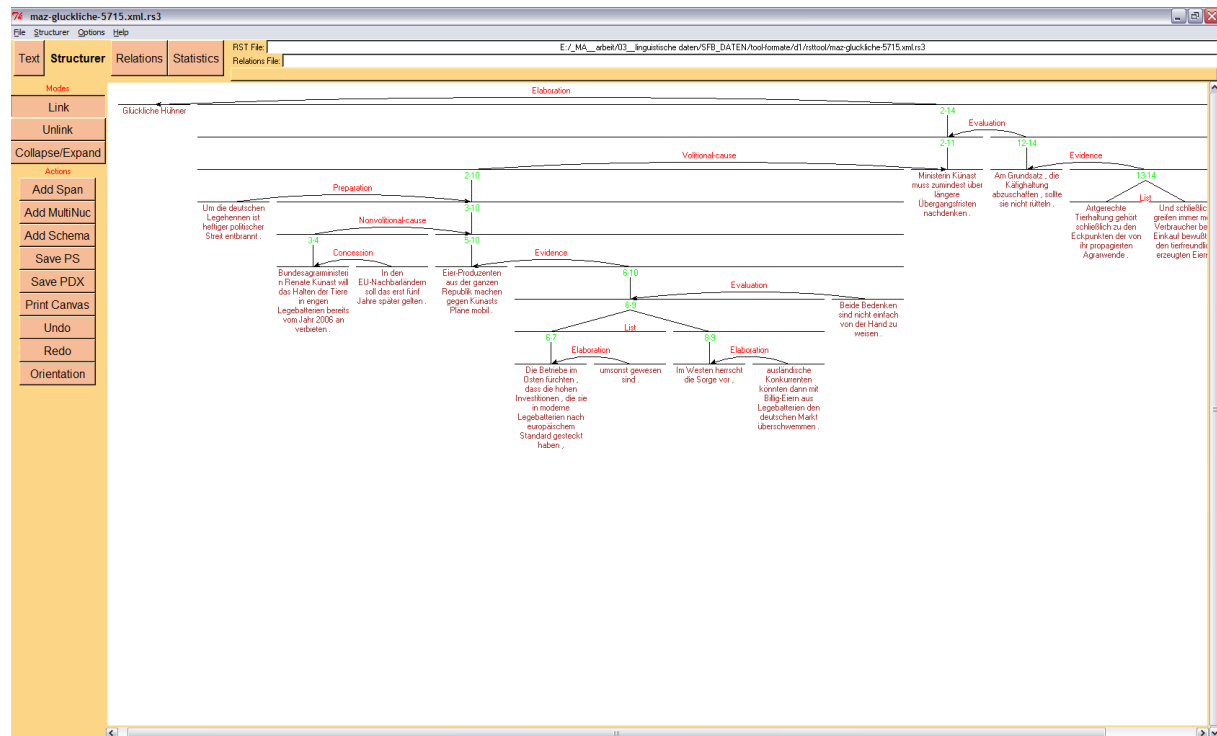


Abbildung 9: Screenshot RSTTool

Umfang und Art der dargestellten Daten: Es können monologische Textdaten und ihre hierarchische Struktur (Rhetorical Structure) dargestellt werden. Die Länge der Texte ist dabei nicht begrenzt. Mehrfache Annotationen zu einem Text innerhalb einer Ansicht/Datei sind nicht möglich.

Datenvisualisierung: Die Visualisierung der Rhetorical Structure orientiert sich an den Visualisierungsrichtlinien, wie sie von Mann & Thompson 1987 detailliert beschrieben wurden. Dabei werden benachbarte Textsegmente in benachbarten Spalten angezeigt und die definierten Beziehungen zwischen den Spalten über Graphen visualisiert. Es wird zwischen Beziehungen unterschieden, in denen die Segmente „gleichberechtigt“ sind (parataktische Beziehungen), und solchen, in denen eine Abhängigkeit zwischen den Segmenten besteht (hypotaktische Beziehungen). Erstere werden über gerade Verbindungslinien visualisiert, letztere über bogenförmige Pfeile.

Interaktion: Der Annotationsprozess basiert im wesentlichen auf der Methode der direk-

³⁰ Mann & Thompson 01.06.1987 (Die Ausführungen sind in der späteren Publikation von 1988 stark verkürzt dargestellt)

ten Manipulation. Nachdem der Text in einem ersten Schritt segmentiert wurde (durch Einfügen von Markierungen in den Fließtext), und die Segmente automatisch als Textspalten nebeneinander angeordnet wurden, kann in einem nächsten Schritt die hierarchische Struktur beschrieben werden. Dies geschieht, indem per *Drag and Drop* eine Beziehung zwischen zwei benachbarten Spalten hergestellt wird, und diese Beziehung durch Auswahl eines Items aus einer automatisch zur Verfügung gestellten Liste charakterisiert wird.

Konfigurationsmöglichkeiten: Textfarbe, Farbe der Graphen und der Nummerierungen sowie des Hintergrundes können angepasst werden. Ebenso können die Spaltenbreite definiert und Schriftfont und -größe eingestellt werden.

Bewertung

Anzahl an darstellbaren Annotationsebenen: Es können keine überlappenden Hierarchien dargestellt werden; damit ist nur eine Annotationsebene visualisierbar (z.B. RST oder Konstituentenstruktur).

Vergleich verschiedener Annotationsebenen: Es wird nur eine Annotationsebene angezeigt.

Heterogenität der Daten: Es können ausschließlich hierarchische Daten visualisiert werden. Welche Hierarchie innerhalb des Textes das ist, spielt dabei keine Rolle; das ist letztlich abhängig von der annotierenden Person und ihrem Forschungsinteresse; einzige Beschränkung ist die untere minimale Segmentgröße (das ist der Buchstabe).

Darstellung längerer Texte: Die Darstellung ist bedingt auch für längere Texte geeignet. Allerdings geht bei längeren Texten die Übersicht über die Struktur verloren, wenn die Baumdarstellung nicht mehr in den Anzeigebereich passt und erst erscrollt werden muss. In dem RSTTool wird die Möglichkeit angeboten, Teile des Baumes zu komprimieren, so dass eine größere Anzahl von in der Hierarchie höhergelegenen Segmenten sichtbar ist. Auch damit ist die Übersicht jedoch eingeschränkt – diesmal nicht in der Breite, sondern in der Tiefe. Eine Kombination mit einem verkleinerten Strukturbaum, in dem der aktuelle Ausschnitt markiert ist, wäre hier evtl. sinnvoll.

Visualisierung von Position: Die Textsegmente selbst sind (auf der Oberfläche) nur indirekt nummeriert: Nummerierungen sind nur den Segmentgruppen zugeordnet, die miteinander verbunden sind (z.B. „3-4“ für eine Gruppe bestehend aus den Segmenten 3 und 4).

Sonstiges: Insgesamt ist die Darstellung sehr raumfordernd. Dies könnte bei der gleichzeitigen Darstellung weiterer Annotationsebenen problematisch sein.

Die Linie, die der Zeilenmarkierung dient, befindet sich auf der gleichen perzeptuellen Ebene wie die eigentlichen Graphen. Die Struktur könnte besser fokussiert werden, wenn

das Raster in den Hintergrund tritt, z.B. heller dargestellt wird.

Die Spalten sind nicht innerhalb einer Zeile ausgerichtet, sondern „springen“ in der vertikalen Ebene. Dadurch wird der Lesefluss behindert.

Referenzen

O'Donnell Juni (2004)

Mann June (1987)

6 TIGERSearch

Kurzbeschreibung

TIGERSearch ist im Rahmen des TIGER Projektes entwickelt worden, um die umfangreichen syntaktischen Daten ³¹der TIGER Treebank explorieren zu können. In dieses Suchtool integriert wurde der so genannte TIGERGraphViewer, mit dem sich Suchergebnisse auf differenzierte Suchanfragen in Bäumen visualisieren lassen.

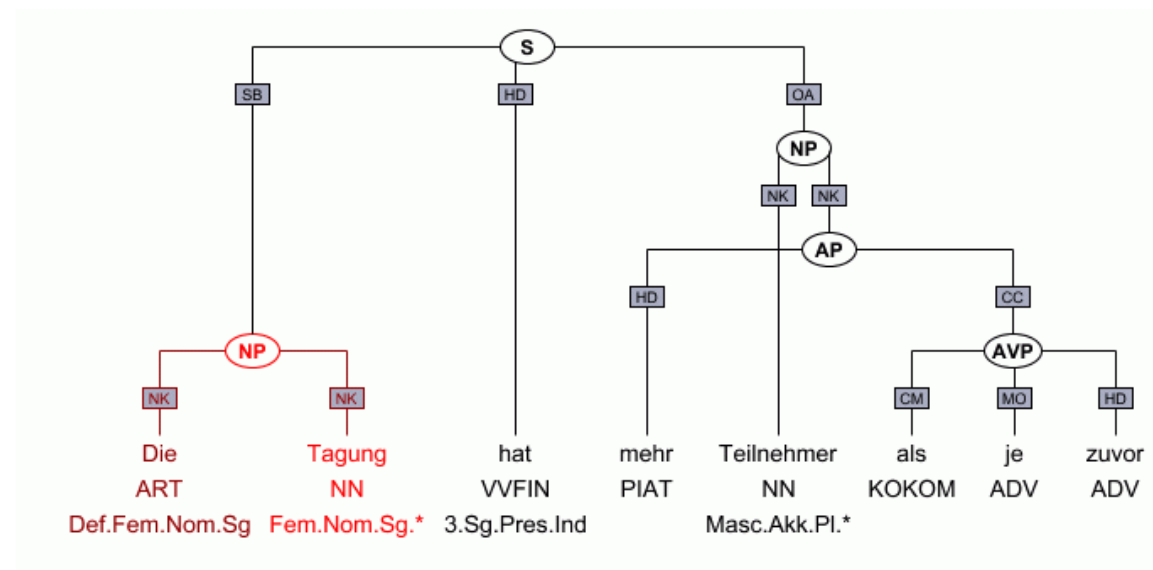


Abbildung 10: Screenshot aus TIGER

Umfang und Art der dargestellten Daten: Es werden mehrere Annotationsebenen dargestellt. Neben hierarchischen Syntaxdaten werden einfache Attribut-Wert-Paare (part-of-speech, Lemmata, oder Übersetzung) als terminale Knoten des Graphen visualisiert.

Datenvisualisierung: Die syntaktischen Annotationen werden in Form eines Graphen visualisiert. Grundform ist ein Baumgraph, der durch eine hierarchische Anordnung seiner Elemente gekennzeichnet ist. Knoten und Kanten des Graphen sind dabei mit den

³¹ Im Jahr 2002 waren es 35.000 syntaktisch annotierte Sätze aus deutschen Zeitungen.

Werten der Annotation bezeichnet. Die Bezeichnungen der Knoten können von den Bezeichnungen der Kanten durch Position und Formatierung unterschieden werden.

Zusätzlich zum reinen Baum werden auch kreuzende Kanten visualisiert, was es ermöglicht, lokale und nicht-lokale Abhängigkeiten zwischen den Elementen darzustellen (als Ersatz für die konventionelle Darstellung der Spuren). Es ist weiterhin möglich, Teilbäume einzeln hervorzuheben.

Interaktion: Es sind vielfältige Möglichkeiten der Interaktion angelegt. Dabei kommen u.a. Techniken der Direkten Manipulation zum Einsatz.

Konfigurationsmöglichkeiten: Es sind vielfältige Möglichkeiten der Konfiguration gegeben: Farben, Hintergründe, Rahmen etc. können konfiguriert werden.

Bewertung

Anzahl an darstellbaren Annotationsebenen: Es können zwar mehrere Annotationsebenen angezeigt werden, jedoch ist – bedingt durch die vertikale Ausdehnung des Graphen – eine gleichzeitige Ansicht vieler Ebenen nicht möglich. Ein unmittelbarer Vergleich zweier Graphen ist nicht möglich.

Vergleich verschiedener Annotationsebenen: Zwischen den eingeblendeten Attribut-Wert-Paaren und den Graphen ist eine visuelle Beziehung gut herzustellen.

Heterogenität der Daten: einfache Attribut-Wert-Paare sowie hierarchische (syntaktische) Strukturen können gut abgebildet werden; für die Darstellung von Koreferenz ist die Visualisierung jedoch nicht geeignet.

Darstellung längerer Texte: Längere Texte können zwar dargestellt, jedoch können Teile des Textes erst nach Scrollen eingesehen werden. Damit ist die Übersicht über den Text nicht gegeben und Exploration von weit verteilten Attributen, beispielsweise von Koreferenzdaten nicht möglich.

Visualisierung von Position: Die Position eines aktuell angezeigten Satzes innerhalb des Gesamttextes wird automatisch angezeigt.

Referenzen:

Brants et al. (2002)

Möglichkeiten der Visualisierung von Text und Annotationen

„Das *Mapping*, die Auswahl einer geeigneten Darstellungsform und die Wahl der graphischen Elemente und Attribute, die zur Darstellung einzelner Datenaspekte verwendet werden, stellt den Kernprozeß der Visualisierung dar“ (Schumann 2000: 125).

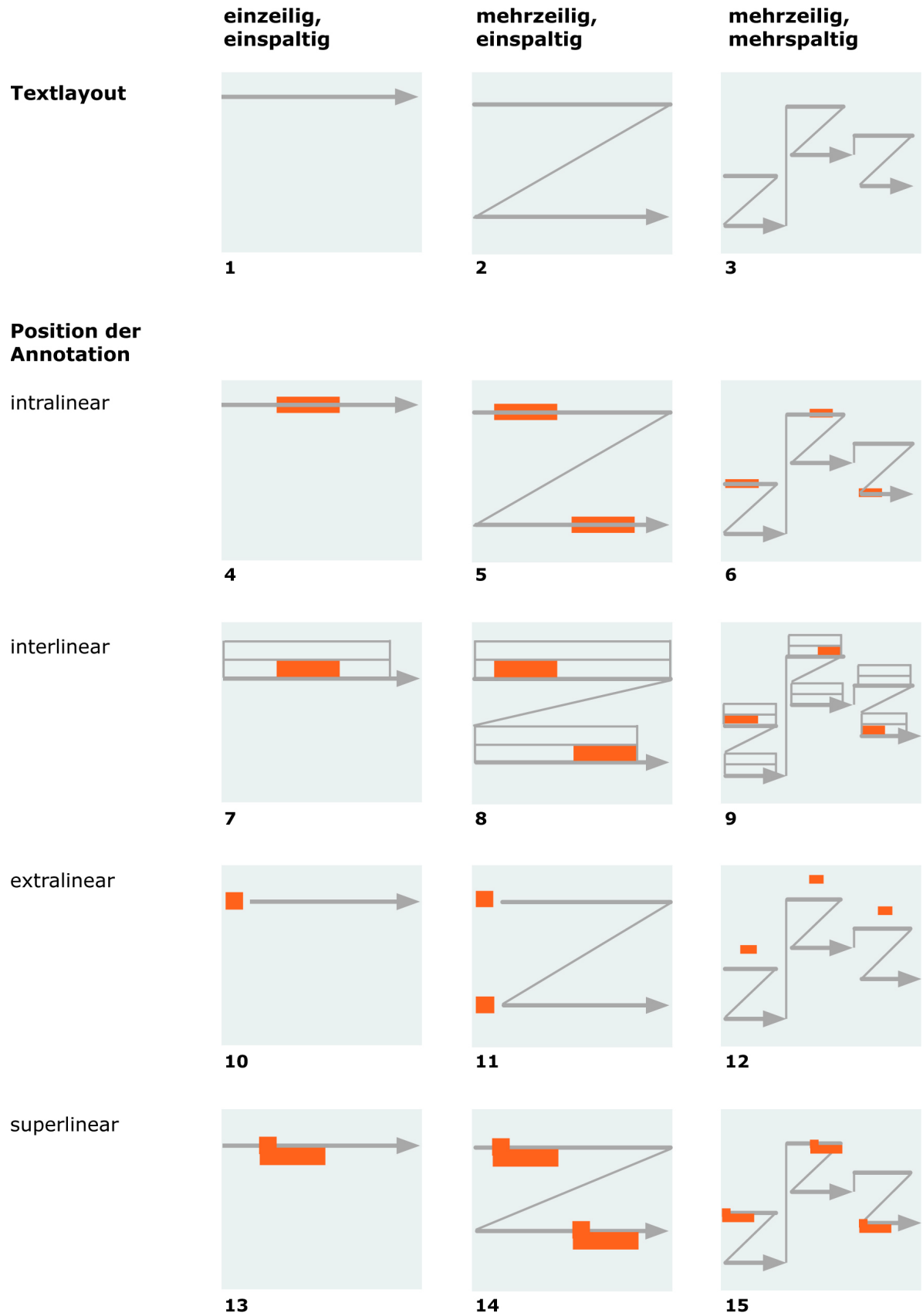
Die Modellierung variabler Visualisierungen für reich annotierte linguistische Daten erfordert nicht nur eine Kenntnis der Nutzer und ihrer Daten, sondern auch eine Kenntnis der verfügbaren grafischen Variablen. Im Folgenden sollen Möglichkeiten der Visualisierung von Primärtext im Zusammenhang mit Annotationen zunächst rein formal betrachtet werden. Es wird das Ziel verfolgt, einzelne Variablen voneinander zu trennen, die die Visualisierungsformen ausmachen. Damit soll eine Basis gelegt werden für ein variables Visualisierungskonzept und eine modulare technische Umsetzung, die es erlaubt, nicht nur während möglicher Nutzertests möglichst viele Parameter getrennt variieren zu können, sondern auch zu einem späteren Zeitpunkt den Nutzern selbst geeignete Einstellungsmöglichkeiten anbieten zu können.

Für die Visualisierung von Daten stehen diverse grafische Mittel wie Position, Farbe etc. zur Verfügung (vgl. Schumann & Müller 2000:128). Dabei stellen die Positionierung von Elementen an erster und die Farbe der Elemente an zweiter Stelle die effektivsten Mittel für die Visualisierung nominaler Daten dar (vgl. ebd.). Hinsichtlich der Visualisierung *linguistischer* Daten (speziell der Primärdaten und Annotationen) sollen die Möglichkeiten der Darstellung im Folgenden spezifiziert werden: Das Grundlayout kann sich entweder (1) am Text, oder (2) an den Annotationen orientieren oder aber (3) beide Komponenten in räumlicher Hinsicht voneinander trennen. Da die Positionierung von Elementen als die effektivste Möglichkeit angesehen wird, diese Elemente zueinander in Beziehung zu setzen (vgl. ebd.) und eine „natürliche“ Abbildung der Daten im allgemeinen als nutzerfreundlicher gilt, soll (1) im Vordergrund meiner Betrachtung stehen.

Ein *Text* ist gekennzeichnet durch eine definierte („natürliche“) Abfolge seiner Elemente. Dagegen befinden sich *Annotationen* per se in keiner Ordnung, können jedoch in diverse künstliche Ordnungen gebracht werden, z.B. in eine alphabetische Abfolge oder in die Zuordnung zu Gruppen, z.B. zu einem Annotationsset (diese Gruppen müssten wiederum irgendwie sortiert sein).³² In diese natürlichen oder künstlichen Ausgangsanordnungen müssen sich die jeweiligen zugehörigen Elemente einfügen. Ich gebe zunächst einen visuellen Überblick über die möglichen Varianten der gemeinsamen Visualisierung von Text und Annotationen und gehe dann detaillierter auf die einzelnen Varianten ein (dies beschränkt

³² bei einer integrierten quantitativen Komponente wäre auch eine Abfolge nach der absoluten/relativen Häufigkeit des Vorkommens wäre denkbar

auf die Visualisierungsformen, in denen der Text das grundlegende Layout bestimmt):



Die Annotationen können zu den Textsegmenten auf verschiedene Arten visuell in Beziehung gesetzt werden. Hinsichtlich der Positionierung der Annotationen lassen sich vier Grundtypen unterscheiden:

- (1) Die Annotationen sind in der gleichen Zeile wie der Text dargestellt (z.B. SAMPA); diese Darstellungsform nenne ich im Folgenden **intralinear**³³.
- (2) Die Annotationen werden parallel zur Textzeile in einer oder mehreren zusätzlichen Zeilen dargestellt (z.B. EXMARaLDA), die sich zwischen den einzelnen Textzeilen befinden; diese Form nenne ich nach Schmidt die **interlineare** Darstellung (Partitur).
- (3) Die Visualisierung der Annotationen befindet sich außerhalb der Textzeilen, steht aber noch in direktem räumlichem Bezug zum Text (z.B. RST); sie soll im Folgenden **extralinear** heißen.
- (4) Bei der **superlinearen** Visualisierung (z.B. Tooltip) werden die Annotationen zu einem Textsegment erst sichtbar, nachdem das Textsegment aktiviert wurde (z.B. bei Mouseover oder Klick); sie sind dann am ausgewählten Textsegment positioniert und können gegebenenfalls die umgebenden Textsegmente überlagern.

Die 4 Grundtypen sollen im Folgenden hinsichtlich der spezifizierten Anforderungen untersucht und bewertet werden.

1 Intralineaare Visualisierung von Annotationen

Frage 1: Wieviele Annotationsebenen können maximal gleichzeitig visualisiert werden?

Um Annotationen innerhalb des Textes zu visualisieren, stehen verschiedene Möglichkeiten zur Verfügung:

- (a) Die Visualisierung der Annotation geschieht durch Einfügen von Zeichen in den Text³⁴. Es kommen dabei theoretisch alle³⁵ Zeichen (Buchstaben und Nichtbuchstaben) in Frage, die dem Zeicheninventar des verwendeten Zeichensatzes entstammen, sowie sämtliche möglichen Kombinationen von Zeichen, die sich vom normalen Text unterscheiden. Praktisch ist die Anzahl begrenzt auf wenige Zeichen, da die Annotationen vom Nutzer ausgewertet werden sollen und dafür ohne großen Lernaufwand memoriert werden müssen.

³³ in Anlehnung an den Begriff der interlinearen Darstellung von Schmidt (2005)

³⁴ Diese Art der Darstellung wird speziell auch in Notationssystemen für die Gesprächstranskription (z.B. GAT) verwendet.

³⁵ (aber natürlich hinsichtlich ihrer Verwendung definierte)

Beispiele: der [ART] Tag [N]
 [[Der Tag]_{NP} [beginnt]_{VP}]

(b) Die Annotationen werden auf die Schriftzeichen abgebildet, entweder über Textauszeichnungen oder über eine Transformation von Klein- in Großbuchstaben (et vice versa). Es stehen in XHTML und CSS diverse Formatierungsmöglichkeiten zur Verfügung:

Beispiele: **fett**, *kursiv*, unterstrichen, überstrichen, **Schriftfarbe**, Schriftgröße, **Schriftfont**

Für die Auszeichnung über die Schriftfarbe stehen 10 Farben zur Verfügung (12 Basisfarben minus Textfarbe und Hintergrundfarbe).

(c) Die Annotationen werden über Hintergrund + Rahmen abgebildet.

Beispiele: **Hintergrundfarbe** Hintergrundmuster | Rahmen |

Der Rahmen kann in den Merkmalen Rahmentyp (einfach, gepunktet, gestrichelt, doppelt), Rahmenstärke, Rahmenfarbe (12 Basisfarben minus Hintergrundfarbe/n) und Rahmenposition (rechts, links, oben, unten) variieren.

(d) Die Annotationen werden über die Textposition abgebildet:

Beispiel: Ich bin ^{hier}. Und manchmal _{hier}.

Diese Art der Visualisierung bietet sich an in Kombination mit einer Änderung der Schriftgröße und kommt als Index-Visualisierung bei der indizierten Klammerung zum Einsatz.

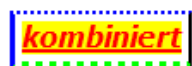
Einschränkung der Möglichkeiten: Annotationen, die in den Text eingefügt sind, sollen vermieden werden, da sie (nach Nutzerurteilen) die Lesbarkeit des Textes zu stark beeinträchtigen. Von den verbleibenden Möglichkeiten sind weiterhin abzuziehen: (a) der Schriftfont, da nicht gewährleistet ist, dass auf jedem lokalen System die erforderliche Schrift installiert ist³⁶, und bei einer Unterscheidung „Serifenschrift versus Nicht-Serifenschrift“ fraglich ist, ob sie für eine schnelle Differenzierung von Segmenten geeignet ist; (b) die Schriftgröße, da eine Veränderung der Zeilenumbrüche durch Verrutschen des Textes möglichst verhindert werden soll; (c) das Hintergrundmuster, das die Lesbarkeit des Textes zu stark beeinflussen würde.

³⁶ Es ist ohnehin fraglich, ob die Unterschiede zwischen verschiedenen Schriftfonts gleicher Größe überhaupt schnell genug wahrgenommen würden.

Dies beschränkt die elementaren Visualisierungsmöglichkeiten auf die folgenden:

visuelle Variable	mögliche Werte	Anzahl der Werte
fett	{fett, nicht fett}	2
kursiv	{kursiv, nicht kursiv}	2
unterstrichen überstrichen kein Strich	{unterstrichen, überstrichen, kein Strich}	3
Schriftfarbe	{rot, grün, gelb, blau, schwarz, weiß, pink, türkis, grau, orange, braun, lila}	10 (12 minus normale Schriftfarbe minus Hintergrundfarbe)
Hintergrundfarbe	{rot, grün, gelb, blau, schwarz, weiß, pink, türkis, grau, orange, braun, lila}	10 (12 minus normale Hintergrundfarbe minus Schriftfarbe)
Rahmen rechts	{ja, nein}	(2)
Rahmen links	{ja, nein}	(2)
Rahmen oben	{ja, nein}	(2)
Rahmen unten	{ja, nein}	(2)
Rahmenstärke	{normal, stark}	2 (bezieht sich auf jede der 4 Rahmenpositionen)
Rahmenfarbe	{rot, grün, gelb, blau, schwarz, weiß, pink, türkis, grau, orange, braun, lila}	10/11 (12 minus Hintergrundfarbe/n) (bezieht sich auf jede der 4 Rahmenpositionen)
Rahmenstil	{normal, gepunktet/gestrichelt, doppelt}	3 (bezieht sich auf jede der 4 Rahmenpositionen)

Da es sich um elementare grafische Variablen handelt, sind diverse Kombinationen möglich. Das folgende Beispiel zeigt folgende Ausprägungen: fett, kursiv, unterstrichen, Schriftfarbe rot, Hintergrundfarbe gelb, Rahmen rechts (schwarz, durchgezogen, dünn), Rahmen links (blau, durchgezogen, dick), Rahmen oben (blau, gepunktet, dünn), Rahmen unten (grün, gepunktet, dick). Es sind darin mindestens **17 Werte** kodiert (wenn man das Vorhandensein der jeweiligen Rahmen nicht als eigene Variable betrachtet):



Die Rahmenstärke ist dabei die einzige visuelle Variable, die quantitativen Charakter hat (abgesehen von der Anzahl der vorhandenen Rahmen) und damit geeignet wäre, quantitative Werte abzubilden (vgl. {Bertin 1974 #87}). Für die Darstellung nominaler Daten mittels der Rahmenstärke ist eine Reduktion auf wenige distinktive Rahmenstärken denkbar – mehr als 2 oder 3 unterschiedliche Werte wären dabei vermutlich nicht mehr praktikabel (dünn, mittel, stark). Wieviele und welche Kombinationen am besten vom Nutzer wahrgenommen wird, ist eine Frage, die für das Festlegen von Defaultwerten beantwortet werden müsste. Zunächst soll es ausreichen, dem Nutzer verschiedene Varianten bieten zu können, die er selbst auswählen kann.

Frage 2: Welche Datentypen können intralinear visualisiert werden? Können sie gleichzeitig

visualisiert werden?

Datentyp	Möglichkeit der Visualisierung	Bewertung/Kommentar
Attribut-Wert-Paare	z.B. Farbe	--
Hierarchie 1 (Satzebene)	indizierte Klammerung, z.B. ()	von Nutzern als störend empfunden
Hierarchie 2 (RST)	indizierte Klammerung, z.B. [], in Abgrenzung zur Satzebene, oder Abgrenzung über andere Farbe der Klammern	von Nutzern als störend empfunden
Attribut-Wert-Gruppen mit/ohne Richtung (z.B. Koreferenz)	z.B. über Hintergrundfarbe und Fettung	als positiv bewertet
Position	z.B. über Indizes oder eingefügte Positionsangaben	--

Frage 3: Eignet sich die Darstellung für kurze und lange Texte? Die intralineare Darstellung ist wegen ihrer Kompaktheit für lange Texte geeignet.

Frage 4: Kann die Darstellung einfach gedruckt werden? Ein Ausdrucken der Darstellung ist unproblematisch. Bei geeigneter Wahl der visuellen Merkmalsausprägungen können sogar Darstellungen gefunden werden, die für Schwarz/Weiß-Drucker funktionieren (gestrichelte vs durchgezogene Rahmen, Hintergrund, fett, kursiv).

2 Interlineare Visualisierung von Annotationen

In diesen Visualisierungsformen bleiben die Textsegmente des Originaltextes ebenfalls in ihrer „natürlichen“ Abfolge. Die Annotationen werden allerdings nicht wie in der intralinen Darstellung auf einer Ebene mit dem Text visualisiert, sondern dazu parallel in eigenen Zeilen.

Diese Darstellungsform, in der Linguistik unter dem Namen „Partitur“ bekannt war ursprünglich ein Notationskonzept in der Musik, erwies sich jedoch mit ihrer Möglichkeit zur Darstellung synchroner Ereignisse nicht nur als hilfreich bei Abbildung von Sprachereignissen. Zunächst wurde sie bei der Transkription gesprochener Sprache (HIAT) eingesetzt, speziell bei der Transkription sich überlappender Sprecherbeiträge. Später wurde von der Gebundenheit an einen realen zeitlichen Verlauf (time-aligned) abstrahiert und das Konzept der Synchronizität insofern erweitert, als nun auch geschriebene Texte ohne unmittelbare zeitliche Fixierung in der Form einer Partitur dargestellt wurden: Als Referenzebene dient hier nicht mehr die temporale/metrische Skala, sondern der schriftlich fixierte Text selbst (element-aligned) (vgl. auch Schmidt 2005:94). Sowohl in der zeitgebundenen als auch in der elementgebunden Variante können – über die Sprecherbeiträge hinausgehend – beliebige Inhalte zur Referenzebene in Beziehung gesetzt werden.

Eine Partiturdarstellung ist auf visueller Ebene gekennzeichnet durch ihre Zweidimensionalität, die sie von einem normalen eindimensionalen Fließtext unterscheidet. Während bei einem einfachen Text die Symbole nacheinander auf einer horizontalen Linie aufgereiht sind, erhält die Partitur in der Vertikalen ihre zweite Dimension. Elemente, die sich auf derselben Vertikalen befinden, sind auf dasselbe Referenzsegment bezogen.

Frage 1: Wieviele Annotationsebenen können maximal gleichzeitig visualisiert werden?

Es können beliebig viele Annotationsebenen parallel zum Text dargestellt werden. Beschränkt ist die Anzahl nur durch die Größe des Anzeigebereichs.

Frage 2: Welche Datentypen können interlinear visualisiert werden? Können sie gleichzeitig visualisiert werden? Es können alle Datentypen interlinear visualisiert werden: einfache Attribut-Wert-Paare, Hierarchie 1 (Syntax), Hierarchie 2 (RST), Attribut-Wert-Gruppen (mit/ohne Richtung) und die Satzposition. Die Visualisierung von Hierarchien und Attribut-Wert-Gruppen wird jedoch von Nutzern als unzureichend gewertet.

Frage 3: Eignet sich die Darstellung für kurze und lange Texte? Mit Einschränkungen. Da immer nur ein relativ kleiner Ausschnitt aus einem längeren Text angezeigt werden kann, ist die Darstellung nicht für die Exploration von Annotationen geeignet, die weit voneinander entfernt liegen.

Frage 4: Kann die Darstellung einfach gedruckt werden? Dies ist abhängig von der Layoutform. Eine einzeilige Darstellung kann nicht automatisch auf Papierformat umgebrochen werden. Für eine Druckfunktion wären zusätzliche Transformationen nötig. Eine mehrzeilige Darstellung, in der also bereits ein Zeilenumbruch erfolgt ist, kann nur dann aus dem Browser heraus ausgedruckt werden, wenn die Zeilenbreite auf das Papierformat passt, ansonsten müsste auch hier zunächst eine neue Tabelle gerendert werden.

3 Extralineare Visualisierung von Annotationen

Frage 1: Wieviele Annotationsebenen können maximal gleichzeitig visualisiert werden?

Bei der extralinearen Visualisierung ist die minimale Größe der Bezugssegmente determiniert durch den Inhalt der Zeile. Das bedeutet: Wenn es sich bei den Segmenten um

RST-Segmente handelt, können nur solche Annotationen in extralinearer Anordnung auf die Zeilen bezogen werden, deren kleinste Bezugseinheit eben dieses RST-Segment ist.

Frage 2: Welche Datentypen können extralinear visualisiert werden? Können sie gleichzeitig visualisiert werden?

Es können alle Daten visualisiert werden, die sich auf das entsprechende Segment beziehen. Insbesondere ist die Darstellung für die Visualisierung von Bäumen geeignet.

Frage 3: Eignet sich die Darstellung für kurze und lange Texte?

Dies ist abhängig von der Position der extralinenaren Annotationen. Werden sie über den Textsegmenten angezeigt, ist ein Zeilenumbruch nicht möglich und damit kann nur wenig Text gleichzeitig angezeigt werden. Befinden sich die Annotation hingegen seitlich vom Primärtext, kann – abhängig jedoch von der Zeilenlänge des Primärtextes – mehr Text angezeigt werden.

Frage 4: Kann die Darstellung einfach gedruckt werden?

In der horizontalen Ausrichtung ist ein Ausdruck bei langen Texten schwierig, in der vertikalen Ausrichtung jedoch gut möglich.

4 Superlineare Visualisierung von Annotationen

Frage 1: Wieviele Annotationsebenen können maximal gleichzeitig visualisiert werden?

In einem überblendenden Bereich können theoretische beliebig viele Annotationsebenen angezeigt werden; praktisch kann es aber von Nachteil sein, wenn die Visualisierungen den Primärtext in großem Umfang überdecken.

Frage 2: Welche Datentypen können superlinear visualisiert werden? Können sie gleichzeitig visualisiert werden?

Es können einfache Attribut-Wert-Paare dargestellt werden.

Frage 3: Eignet sich die Darstellung für kurze und lange Texte?

Hinsichtlich der Textlänge ist kein negativer Effekt zu erwarten

Frage 4: Kann die Darstellung einfach gedruckt werden?

Die Darstellung kann nicht gedruckt werden.

5 Zusammenfassung

Die Visualisierung der Annotationen kann zeitgleich zum Text erfolgen oder aber zeitlich versetzt als Reaktion auf eine Nutzeraktion hin. Während bei den Typen intralinear,

interlinear und extralinear eine gleichzeitige Visualisierung von Annotationen und Text möglich ist, ist dies bei der superlinearen Variante wegen möglicher Überlagerung nur beschränkt auf ein bestimmtes Textsegment möglich. Hier bietet sich eine zeitliche Versetzung an. Aber auch für die ersten drei Fälle ist eine zeitliche Versetzung denkbar und sinnvoll: Da nicht zu jedem Zeitpunkt während der Exploration alle verfügbaren Annotationen von Interesse sind, sollte hier die Möglichkeit bestehen, Annotationen erst bei Bedarf zur vorhandenen Visualisierung hinzuzufügen.

Übersicht über Vor- und Nachteile der Visualisierungsformen

- (1) **intralinear**: Annotationen werden zeitgleich zum Text innerhalb des Textes visualisiert (z.B. mit Schriftfarbe oder Hintergrundfarbe)
Vorteile:
 1. Die Darstellung ist kompakt und damit gut für längere Texte geeignet.
 2. Die Darstellung ist gut geeignet für die Darstellung von Koreferenzketten.*Nachteile:*
 1. Es ergibt sich die Schwierigkeit, sich überschneidende Annotationen gleichzeitig zu visualisieren.
 2. Die Darstellung ist nicht für beliebig viele verschiedene Annotationen geeignet.
 3. Hierarchien können nur mit indizierter Klammerung abgebildet werden, was bei komplexen Hierarchien leicht unübersichtlich wird (gesplittete Elemente sind über nummerierte Indizes visualisierbar).
- (2) **interlinear (Partitur)**: Annotationen werden auf zusätzlichen Zeilen visualisiert, die parallel zum Text verlaufen und sich auf die Textzeile beziehen
Vorteile:
 1. Es können viele Ebenen zueinander in Beziehung gesetzt werden.
 2. Die Segmente der Annotationsebenen können sich überschneiden*Nachteil:*

Die Visualisierung ist raumfordernd.
- (3) **extralinear**: die Visualisierung der Annotationen befindet sich außerhalb der Zeilen, aber in direkter räumlicher Beziehung zu ihnen (z.B. bei vertikaler RST-Version)
Vorteil:

Es ist eine Darstellung von Hierarchien möglich, deren Bezugssegmente längere Textsegmente sind. Wenn die extralinear Darstellung sich seitlich von diesen Segmenten befindet, kann eine kompakte Darstellungen der Hierarchie erfolgen. In der Vertikalen wäre ebenfalls ein Vergleich zweier hierarchischer Strukturen möglich.

Nachteil:

Eine horizontale Darstellung ist stark raumfordernd.
- (4) **superlinear**: die Visualisierung der Annotationen wird über den Zeilen eingeblendet.
Vorteil:

Ein Einblenden der Annotationen geschieht direkt am (mit der Maus) fokussierten Textsegment.

Nachteil:

Wenn die superlineare Darstellung mehrere Annotationsebenen visualisiert, kann nicht der Geltungsbereich für alle Attribut-Wert-Paare gleichzeitig angezeigt werden. Das bedeutet, dass der Nutzer nicht weiß, auf welche Textsegment sich die eingeblendeten Annotationen beziehen.

Konzept für die Visualisierung reich annotierter linguistischer Daten

1 Guidelines

1 Farben

Die visuelle Variable *Farbe* stellt nach der Variablen *Position* die zweiteffektivste Möglichkeit dar, nominale Daten zu visualisieren. Über die Effektivität schreibt Ware (2004: 123): „Color can be extremely effective as a nominal code. When we wish to make it easy for someone to classify visual objects into separate categories, giving the objects distinctive colors is often the best solution“.

An den effektiven Einsatz von Farbe als Variable für die Darstellung nominaler Daten ist eine Bedingung geknüpft: Die eingesetzten Farben müssen distinktiv sein. Aufgrund unterschiedlicher Arbeitsumgebungen des Nutzers mit unterschiedlichen Bildschirmen (TFT, Röhrenbildschirm), die unterschiedlich kalibriert sein können, kann jedoch nicht davon ausgegangen werden, dass kleine Farbdifferenzen von jedem Nutzer sichtbar sind.³⁷ Colin Ware (2004) formuliert auf der Basis von Forschungsergebnissen zur Wahrnehmung von Farbe Regeln, die beim Einsatz von Farbe als visuellem Code beachtet werden sollten (Ware 2004:143-144)³⁸. Es sei an dieser Stelle eine Auswahl zitiert (Übersetzung d. Verf.):

- Wenn Farbe als distinkter Code eingesetzt wird, sollten nur wenige Farben genutzt werden. Sechs distinkte Farben zu wählen, ist einfach; wenn allerdings 10 Farben benötigt werden, sind diese mit Sorgfalt auszuwählen. Wenn der Hintergrund variiert, dann wird der Einsatz von mehr als 12 Farben voraussichtlich zu fehlerhaften Farbbewertungen führen.
- Schwarze oder weiße Rahmen um die farbigen Symbole können helfen, diese distinkt zu machen, indem der Luminanzkontrast zu den umgebenden Farben unterbrochen wird.
- Rot, Grün, Gelb und Blau sind als Primärfarben fest im Gehirn verankert (wörtlich: verdrahtet). Wenn es notwendig ist, eine Farbkodierung zu erinnern, sind dies die ersten Farben, die erwägt werden sollten.
- Um größere Bereiche farblich zu kodieren, sollten gedeckte Farben eingesetzt werden, ganz besonders, wenn farbige Symbole davor gesetzt werden sollen.
- Kleine farbkodierte Bereiche sollten Farben mit hoher Sättigung erhalten.

Farbe in Hintergrund und Vordergrund

Da Farben abhängig von ihrem farblichen Umfeld wahrgenommen werden und es beim Einsatz von opponierenden Farben (z.B. Blau und Rot) sogar zu Farbkontrastillusionen³⁹ (*color contrast illusion*; ebd.:117) kommen kann, besteht die Gefahr, dass die farbkodierten Daten aufgrund ihrer Umgebung falsch bewertet werden. Es ist ein Weg zu finden, wie solche Fehleinschätzungen vermieden werden können.

³⁷ Hinzu können – nicht ganz ungewöhnlich – Farbsehstörungen kommen

³⁸ Im Rahmen dieser Arbeit kann nicht im Detail auf die Farbtheorie eingegangen werden. Es sei an dieser Stelle auf die vielfältige Fachliteratur verwiesen.

³⁹ Definition: ...

Eine Möglichkeit, solche Wahrnehmungsprobleme zu verhindern, wäre, auf den Einsatz von opponierenden Farben ganz zu verzichten, was jedoch die ohnehin geringe Anzahl an distinktiven Farben zu stark reduzieren würde. Stattdessen kann auf die viel einfachere Möglichkeit zurückgegriffen werden, den Hintergrund und die Umgebung der farbkodierten Daten möglichst neutral zu gestalten, damit die farbkodierten Daten unverfälscht wahrgenommen werden. Farbneutral heißt dabei achromatisch, also auf der Grauskala liegend. Damit sich die Farben stark genug vom Hintergrund abheben, wird außerdem ein starker Helligkeitskontrast relativ zum Vordergrund benötigt. D.h. von der Grauskala bleiben: Weiß/helles Grau oder Schwarz/dunkles Grau.

Folgende Farben sind nach (Ware 2004) distinkt: rot, grün, gelb, blau, schwarz, weiß, pink, türkis, grau, orange, braun und lila. Dabei sind weniger gesättigte Farben für den Hintergrund und gesättigte Farben für den Vordergrund zu wählen. Eine mögliche Palette sei im Folgenden definiert.

Farbe		Farbwert (hexadezimal)
rot	dunkel	#ff7246
	hell	#f9d7ce
grün	dunkel	#98cf15
	hell	#e2eec6
gelb	dunkel	#fbe201
	hell	#fcf6cc
blau	dunkel	#a1a9ec
	hell	#e4e7fc
schwarz	dunkel	#000000
	hell	-
weiß	dunkel	-
	hell	-#ffffff
pink	dunkel	#ff62f5
	hell	#fbe2fa
türkis	dunkel	#75d5d0
	hell	#daf1f1
grau	dunkel	a3a3a3
	hell	#e6e6e6
orange	dunkel	#ffb718
	hell	#f9eac8
braun	dunkel	#c59b62
	hell	#ede4d7
lila	dunkel	#d0a6d7
	hell	#f2e1f5

2 Maßangaben

Für die Definition von numerischen Angaben, darunter den Schriftgrößen, stehen in CSS diverse Maßeinheiten zur Verfügung: pt, pc, in, mm, cm, px, em, ex, %. Dabei werden relative und absolute Einheiten unterschieden (vgl. Bos 1998:43ff.).

Relative Größenangaben werden relativ zu einer anderen Größe errechnet. Das hat den Vorteil, dass Bereiche, die in einer relativen Maßeinheit definiert wurden, automatisch angepasst werden, wenn sich die Größe des anderen Elementes verändert; z.B. kann dadurch die Breite eines Bereiches an die Bildschirmgröße bzw. Browsergröße angepasst werden (z.B. bei Größenangaben in %). Wenn Bereiche abhängig von der gewählten Schriftgröße definiert werden (mit der Größenangabe **em**), werden diese automatisch mit vergrößert, wenn der Nutzer im Browser eine größere Schrift einstellt. (vgl.ebd.: 44) weist außerdem darauf hin, dass Stylesheets mit relativen Maßeinheiten einfacher von einem Medium zum anderen skalieren: „Style sheets that use relative units will more easily scale from one medium to another (e.g., from a computer display to a laser printer)“ (Bos 1998 :44). Damit die Visualisierungen sich an die Benutzerumgebung, die persönlichen Schriftgrößeneinstellungen und darüber hinaus auch an mögliche andere Nutzungskontexte (andere Datenbanken) anpassen können, wird hier soweit wie möglich auf die relative Maßeinheit **em** zurückgegriffen.

2 Kontext: User Interface zur Exploration linguistischer Daten

Die Nutzung von Visualisierungen linguistischer Daten erfolgt im Rahmen einer Datenbankabfrage. Die Visualisierungen sind also eingebettet in eine umgebende Nutzeroberfläche, in der z.B. die Suche realisiert wird.

Abbildung 11 zeigt eine schematische Übersicht über ein solches Interface. Es orientiert sich an den von Shneiderman formulierten Phasen bei der Informationssuche „Overview first, zoom and filter, then details on demand“ {Shneiderman 2005 #5: 581}.

Die nachfolgenden Ausführungen zu den Visualisierungen beziehen sich ausnahmslos auf den Darstellungsbereich von Such**ergebnissen** (die rechte Seite in der Abbildung). Ich werde zunächst solche Visualisierungen betrachten, die sich in die **Detailansicht** einfügen. Im Anschluss gehe ich kurz auf erweiterte Formen von **Ergebnislisten** ein, die für den spezifischen Anwendungsfall des Projektes B4 von Relevanz sind.

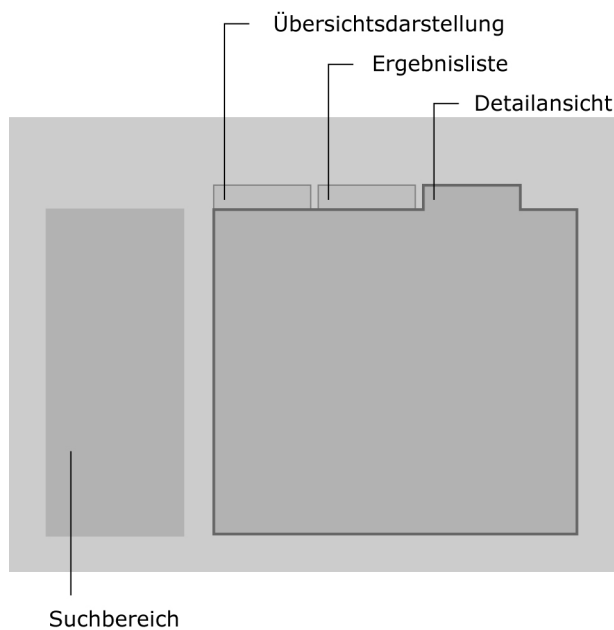


Abbildung 11: schematischer Aufbau einer Nutzeroberfläche

3 kombinierte vs. integrierte Darstellung

Die verschiedenen Annotationen und entsprechenden Rohdaten in vielfältiger Weise zueinander in Beziehung zu setzen, ist als Ausgangspunkt für die Erstellung von Visualisierungen definiert worden. Bisher wurde auf diverse Möglichkeiten eingegangen, wie einzelne Visualisierungstypen für verschiedene Datentypen genutzt werden können. Die Art und Weise der *gleichzeitigen* Darstellung heterogener Daten ist dabei bisher nicht betrachtet worden. Dies soll im Folgenden geschehen.

Um Visualisierungen miteinander in Beziehung zu setzen, sind drei Möglichkeiten vorstellbar:

- (1) Die Visualisierungen können nacheinander angezeigt werden. Dies hat den Vorteil, dass eine Visualisierung den gesamten zur Verfügung stehenden Anzeigebereich einnehmen kann. Das bedeutet allerdings, dass der Nutzer die Ausgangsvisualisierung bzw. spezifische dargestellte Werte memorieren muss, will er diese mit der neuen Visualisierung oder ihren dargestellten Werten in Beziehung setzen. Wenn die Seite oder die Visualisierung erst neu geladen werden muss, kann die Zeitspanne für die Exploration zu groß werden. Möglicherweise wäre diesem Nachteil zu begegnen, indem die Geschwindigkeit des Bildwechsels auf ein Minimum reduziert wird, z.B. indem der Wechsel über ein Mouseover initiiert wird.
- (2) Die Visualisierungen können in einem Anzeigebereich kombiniert dargestellt werden.
- (3) Die Visualisierungen können miteinander verschränkt werden.

Kombinierte Darstellung

Die Daten werden in mehreren Bereichen dargestellt, die jeweils die Visualisierung eines bestimmten Datentyps fokussieren. Die dargestellten Inhalte dieser Bereiche können entweder voneinander unabhängig sein oder aber voneinander abhängig, und zwar insofern als eine nutzerinitiierte Änderung innerhalb des einen Bereichs den Inhalt eines anderen Bereichs beeinflusst (vgl. auch North, Shneiderman 15.10.1998).

Beispiele: (1) Text und Annotationen sind in zwei getrennten Bereichen angesiedelt. Bei Klick auf ein Token in einem Fließtext werden die zugehörigen Annotationen im Annotationsbereich angezeigt. (2) RST und Syntax werden in zwei getrennten Bereichen dargestellt. Bei Klick auf ein Segment in einem der Bereiche (z.B. ein RST-Segment) wird dieses Segment hervorgehoben und gleichzeitig auch das zugehörige Segment im anderen Bereich hervorgehoben. (3) Zusätzlich zu (1) kann im Annotationsbereich ein Wert aus einer Liste verfügbarer Annotationswerte ausgewählt werden, mit der Folge, dass im Textbereich alle Textsegmente markiert werden, die hinsichtlich dieses Wertes annotiert sind.

Problem: Die Existenz mehrerer Fenster kann – in Abhängigkeit von der Bildschirmgröße und der Anzahl der eingeblendeten Fenster – den Überblick über die zu explorierenden Daten stark einschränken. Durch eine variable Größengestaltung der Bereiche und die Möglichkeit, die Fenster ein/auszublenden bzw. in ihrer Position zu verändern, kann hier Abhilfe geschaffen werden.

Bewertung: Kombinierte Darstellungen sind denkbar sowohl für die Darstellung heterogener Daten (1 Bereich pro Datentyp) als auch für die Darstellung konfligierender Annotationen (z.B. Mehrfachannotationen; ein Bereich pro Annotation, inkl. Bezugstext).

Integrierte Darstellung

In dieser Darstellungsform werden zwei oder mehr Visualisierungstypen in einem einzigen Bereich miteinander verschränkt (z.B. RST und Koreferenz). Eine Verschränkung ist nicht für alle Darstellungstypen möglich. Bei der integrierten Darstellung kann auf engem Raum eine Vielzahl an Daten abgebildet werden.

4 Basisvisualisierung: intralineare Darstellung

Auf die Möglichkeiten der intralinearen Darstellung von Attribut-Wert-Paaren wurde bereits ausführlich eingegangen. An dieser Stelle werden lediglich die Probleme bei der Visualisierung von Position dargestellt.

Intralineare Darstellung von Position: Die Position kann über eingefügte Indizes (z.B. bei Token) oder über in den Text eingefügte Nummern visualisiert werden (z.B. bei Sätzen). Diese Art der Visualisierung ist allerdings nur möglich, wenn es auf der gewählten Annotationsebene keine Leerstellen gibt, das heißt, wenn zu jedem Segment der Annotationsebene ein Wert vorhanden ist. Andernfalls kommt es zu einer verfälschenden Darstellung:

Beispiel: Visualisierung von Satzposition

[1] Glückliche Hühner [2] Um die deutschen Legehennen ist heftiger politischer Streit entbrannt . [3]
Bundesagrarministerin Renate Künast will das Halten der Tiere in engen Legebatterien bereits vom Jahr
2006 an verbieten .

Beispiel: *Visualisierung von Informationsstruktur

[1] Glückliche Hühner Um [2] die deutschen Legehennen ist [3] heftiger politischer Streit entbrannt .

Diese Darstellung impliziert die Existenz der drei Segmente „Glückliche Hühner Um“, „die deutschen Legehennen ist“, „heftiger politischer Streit entbrannt“.

Durch zusätzliche Markierung des Bezugssegmentes ließe sich dies jedoch korrigieren:

Beispiel: Visualisierung von Informationsstruktur

[1] Glückliche Hühner Um [2] die deutschen Legehennen ist [3] heftiger politischer Streit
entbrannt .

oder:

[1] Glückliche Hühner Um [2] die deutschen Legehennen ist [3] heftiger politischer Streit
entbrannt .

5 Basisvisualisierung: interlineare Darstellung (Partitur)

Bei der interlinearen Darstellung werden die Beschreibungen der Textsegmente nicht direkt im Text dargestellt, sondern befinden sich in Extra-Zeilen, die sich auf die Textebene beziehen und visuell an diese gekoppelt sind. Die lineare Textzeile ist sozusagen in der Vertikalen expandiert.

Die interlinearen Visualisierungen von ANNIS, EXMARaLDA (und z.B. auch Toolbox) unterscheiden sich im Wesentlichen durch ihre Formatierung. Der Einfluss der Formatierung auf die Perzeption der Daten wird beim Vergleich von EXMARaLDA mit ANNIS augenfällig. Die den realisierten Visualisierungen zugrunde liegende Struktur ist im wesentlichen dieselbe. Ein wesentlicher Unterschied liegt jedoch in der Segmentierung vor:

Während in EXMARaLDA verbundene Segmente dargestellt werden können, ist dies in ANNIS nicht möglich, weshalb die Daten, die sich über zwei Segmente erstrecken nur schwer lesbar sind. Die interlineare Darstellung sollte also in jedem Fall verbundene Segmente ermöglichen.

Das visuelle Bild einer Partitur lässt sich auf struktureller Ebene als Tabelle beschreiben. Diese Tabelle hat eine bestimmte Anzahl an Spalten, die in einer Referenzzeile festgelegt sind. Sie stellen das grundlegende Raster dar. Eine Zelle dieser Referenzzeile ist für die Tabellenstruktur eine atomare Einheit. Die Breite einer Zelle auf einer nachfolgenden Zeile lässt sich beschreiben als horizontale Ausdehnung über eine Anzahl x atomarer Spalten. Da die atomare Einheit der ANNIS-Daten auf den Buchstaben festgelegt ist, wird im Folgenden vom Buchstaben als kleinstmöglicher Spalte ausgegangen. Eine zeitlich fixierte Darstellung schließe ich damit aus meinem Konzept aus.⁴⁰

Im Folgenden ist die grundlegende Struktur für eine zu erstellende Tabelle dargestellt:

<i>Labelspalte</i>	<i>Aktionsspalte</i>	<i>Datenspalte</i>	<i>Datenspalte</i>	<i>Datenspalte</i>	
Tabellennr.		Segmentnr.	Segmentnr.	Segmentnr.	<i>Nummerierung</i>
Label	Aktionen	Bezugstext	Bezugstext	Bezugstext	<i>Referenzzeile</i>
Label	Aktionen	Annotation	Annotation	Annotation	<i>Datenzeile</i>
					<i>Leerzeile</i>
Label	Aktionen	(leer)	Annotation		<i>Datenzeile</i>
Label	Aktionen	Annotation		Annotation	<i>Datenzeile</i>

Es wird unterschieden zwischen einem Datenbereich, in dem der Primärtext und die annotierten Daten enthalten sind, der Zeilen- und Spaltenbeschriftung sowie einem

⁴⁰ Für eine Darstellung, die sich an der Zeitskala orientiert, gelten ohnehin andere Anforderungen als an eine einfache elementgebundene Darstellung. Diese wären zunächst zu spezifizieren.

zusätzlichen (optionalen) Bereich, der Elemente für die Interaktion mit der Tabelle beinhalten kann.

Nach Funktion und Inhalt werden folgende Zeilentypen unterschieden:

- (1) Zeile mit Tabellenummer und Segmentnummern (Anzahl: 1-2): Um die Position eines Elementes überprüfen zu können, werden für den Datenbereich Segmentnummern angezeigt. Jede Spalte des Datenbereichs erhält eine Nummer. Eine Nummerierungszeile steht obligatorisch als erste Zeile in der Tabelle; fakultativ kann eine zusätzliche Nummerierungszeile als letzte Tabellenzeile eingeblendet werden. Außerdem kann es wichtig sein, eine Tabellenummer anzuzeigen, nämlich dann, wenn es mehrere Zeilen für eine Partitur gibt. Diese Tabellenummer wird obligatorisch angezeigt, wenn die Nummerierungszeile die erste Zeile ist. Wenn die Nummerierungszeile die letzte Zeile ist, sollte die Tabellenummer nicht sichtbar sein (also gar nicht erst im XHTML-Code vorhanden, oder aber einfach nicht angezeigt).
- (2) Referenzzeile (Anzahl: 1-n): Die Referenzzeile ist die Annotationsebene, die für die Segmentierung der Tabelle ausgewählt wurde. Im Regelfall ist das die Ebene mit dem Primärtext (in ANNIS die Annotationsebene „words“, die die Token beinhaltet); Segmentierungen unterhalb der Tokenebene sollen möglich sein. Segmentierungen oberhalb der Tokenebene scheinen im gegebenen Kontext nicht nötig zu sein.⁴¹
- (3) Leerzeile (Anzahl: 1-n):
- (4) Datenzeile (Anzahl: n): pro Annotationsebenen können eine oder mehrere Datenzeilen vorhanden sein. Datenzeilen können (a) Einzelzeilen sein: diese sind ausschließlich abhängig von der Referenzzeile oder (b) Bestandteil einer Zeilengruppe sein: die Gruppe kann entweder willkürlich erstellt sein (z.B. projektspezifisch oder als zugehörig zu einem bestimmten linguistischen Bereich) oder durch die Daten selbst obligatorisch sein (z.B. hierarchische Daten). Zeilengruppen können in der Abfolge ihrer Zeilen variabel sein oder festgelegt (z.B. bei Syntax).

Es werden drei Spaltentypen unterschieden: die Labelspalte, die Aktionsspalte und die Datenspalte:

- (1) **Labelspalte:** Die Labelspalte ist obligatorisch; sie beinhaltet den Namen der Annotationsebene. Dieser Name kann entweder in Langform oder in Kurzform angezeigt werden. Ein Umschalten in den jeweils anderen Modus sollte über ein

⁴¹ Es wurde in den Nutzerbefragungen jedoch auch nicht explizit danach gefragt

Konfigurationsmenü möglich sein. Wenn die Kurzform ausgewählt ist, sollte die Anzeige der Langform bei Mouseover verzögert eingeblendet werden (in XHTML über das <title>-Tag zu realisieren).

- (2) **Aktionsspalte:** Die Aktionsspalte ist optional; sie beinhaltet Schaltflächen für Nutzeraktionen, die sich auf eine Zeile beziehen (z.B. Zeile nach oben verschieben, Zeile nach unten verschieben). Die Aktionsspalte wird dem Nutzer nur dann zur Verfügung gestellt, wenn sein Client in der Lage ist, die Aktionen auszuführen. Wenn die Aktionen möglich sind, stehen die Schaltflächen zur Verfügung, können aber bei Bedarf vom Nutzer ausgeblendet werden.
- (3) **Datenspalten:** Die Datenspalten enthalten je nach Zeile die Zellen mit den eigentlichen zu explorierenden Daten. Die Zellen sind minimal so groß wie die Zellen der Referenzzeile und maximal so groß wie die Breite des Datenbereichs. Leere Zellen in den Datenspalten sind gekennzeichnet durch das Fehlen eines Attributwertes. Ihre Segmentierung entspricht der durch die Referenzzeile vorgegebenen Segmentierung. Die so entstehende durchgehende vertikale Segmentierung lässt verbundene Zellen besser hervortreten. Sonderfall ist die Leerzeile: Hier werden alle Datenzellen miteinander verbunden, damit ein durchgängiger Abstand zwischen den Gruppen geschaffen wird.

Problembereiche:

Folgende Problembereiche sollen geklärt werden:

- (1) Kontextualisierung: Wie kann erreicht werden, dass der Kontext eines Partiturausschnittes unmittelbar zur Verfügung steht?
- (2) Wie kann die Lesbarkeit bei einer großen Anzahl von Annotationsebenen erhalten bleiben? (Prinzip: viel auf wenig Raum)
- (3) Wie können flexibel Annotationsebenen zueinander in Beziehung gesetzt werden?
- (4) Wie können beschreibende Zusatzinformationen (z.B. zum Tagset oder zum Attributwert) verfügbar gemacht werden?

Kontextualisierung

Die interlineare Darstellung kann – abhängig von der Anzahl der Annotationsebenen und von der Länge des Primärtextes größer sein als der Anzeigebereich. Dies bedeutet für den Nutzer, dass er nur einen Teil der vorhandenen Daten einsehen kann.

Einzelne kurze Sätze (wie im Projekt B4) können ohne Probleme in einer Partitur/Tabelle

innerhalb des limitierenden Anzeigebereiches dargestellt werden. Längere Sätze, die breiter sind als ihr Container, können nicht vollständig angezeigt werden, müssen aber irgendwie dem Nutzer zur Verfügung gestellt werden. Dafür sind verschiedene Lösungen denkbar: (a) Vor/Zurück-Navigation bezogen auf die Zeichen des Satzes oder eine bestimmte Anzahl von Segmenten (diese Möglichkeit wurde in ANNIS gewählt, scheint aber nicht praktikabel), (b) Auswahl eines Textausschnittes durch Klick auf einen Übersichtstext (auch diese Lösung wurde in ANNIS umgesetzt und wird von den Nutzern als sinnvoll beurteilt, wenn längere Texte exploriert werden sollen), (c) horizontale Scrollbalken, über den der verborgene Teil der Darstellung schnell zur Verfügung gestellt werden kann, ohne dass der Nutzer dafür seine Aufmerksamkeit auf eine externe Navigation lenken müsste, (d) Umbruch der Partiturzeile – es bleibt soviel Partitur in einer Zeile wie das Containerelement breit ist, der restliche Teil der Partitur wird in die nächste Zeile umgebrochen. Bei längeren Texten oder vielen Annotationsebenen wird diese Darstellung in der Vertikalen scrollbar. Diese Möglichkeit wäre leicht zu drucken. Ein Problem ist allerdings die visuelle und technische Realisierung der Bruchstellen.

Es sollen eine einfache scrollbare Partitur, eine navigierbare Partitur und – wenn möglich – eine umgebrochene Partitur realisiert werden.

Lesbarkeit

Die Lesbarkeit bei einer großen Anzahl von Ebenen kann verbessert werden durch eine visuelle Clusterung von Zeilen zu Zeilengruppen. Dies geschieht durch das Hinzufügen von Leerzeilen an eine beliebige Stelle innerhalb der Partitur.

Vergleich von Annotationsebenen

Der Vergleich zwei entfernten Annotationsebenen kann verbessert werden, indem (a) ein Hervorheben von Zeilen ermöglicht wird (temporär durch Mouseover oder dauerhaft durch Klick auf das Zeilenlabel) und (b) die Zeilen in direkte Nachbarschaft verschoben werden können (durch Klick auf Pfeile in der Aktionsspalte)

Der Zusammenhang von *Werten* auf zwei entfernten Annotationsebenen kann verbessert werden durch Markierung dieser Werte; bei Klick auf einen Wert erhält der Nutzer die Möglichkeit zu wählen, nur diesen einen Wert anzuzeigen oder alle Zellen mit diesem wert.

Zusatzinformationen

Abkürzungen sollten an der Stelle aufzulösen sein, an denen die Frage nach ihrer Bedeutung entsteht. Die Auflösung von Abkürzungen ist eher für Nutzer relevant, die nicht mit einer Annotation der Daten beschäftigt waren und die Abkürzungen nicht unmittelbar auflösen können.

[1]	1	2	3	4	5	6	7	8
words	Um	die	deutschen	Legehennen	ist	heftiger	politischer	Streit
lemma	um	der	deutsch	Legehenne	sein	heftig	politisch	Streit
POS	AP	ART	ADJA	NN	VAF	ADJA	ADJA	NN
cs 3	S							
cs 2	NP(1)				NP(2)			
cs 1	PP							

Diese Variante ist geeignet, eine große Anzahl von Annotationsebenen untereinander darzustellen.

Neuerungen gegenüber ANNIS

- Zeilen sind benannt (und fixiert)
- Durch die größere räumliche Nähe der Annotationssets rücken die Ebenen näher zusammen und erleichtern ein schnelles Erfassen der Daten .
- Der rechte und linke Kontext des aktuellen Partiturausschnittes sind ohne erneutes Laden der Seite einsehbar.
- Die Nummerierung orientiert sich an der segmentbestimmenden Zeile (hier: token)
- Nachteil: Die Zugehörigkeit der einzelnen Ebenen zu einem definierten Annotationsset ist nicht mehr unmittelbar ersichtlich; auch wenn diese Information im Vergleich zu so grundlegenden Informationen wie dem Zeilennamen sekundär ist, sollte sie (z.B. in einem kleinen Info-Fenster) verfügbar gemacht werden.

Der Entwurf wurde von den Nutzern als positiv gewertet und als vorteilhafter gegenüber der Darstellung in EXMARaLDA; dabei ist die interaktive Komponente noch nicht betrachtet worden (die Beurteilungen erfolgten bezogen auf einen (Farb-)Ausdruck).

mögliche Probleme: Konzept der Zeilengruppe

Ein besonderes Problem stellt evtl. das **Konzept der Zeilengruppe** dar: Da auf der Oberfläche nicht ersichtlich ist, ob es sich um eine Zeilengruppe handelt oder nicht, könnte es zu Verwirrung führen, wenn sich einige Zeilen nur als Komplex verschieben lassen, andere hingegen einzeln.

Es ist zu prüfen, ob das für den Nutzer ein reales Problem darstellt: Wenn ja, müsste versucht werden, das Konzept der Zeilengruppe extra zu visualisieren, möglicherweise durch ein Verknüpfungssymbol wie in Photoshop oder durch eine Art eckige Klammer in Labelnähe (nicht jedoch am Zeilenende, da hier evtl. Linien für die Darstellung des Zeilenumbruchs benötigt werden).

Stilvarianten

(1)

[1]	1	2	3	4	5	6	7	8	9
words	Um	die	deutschen	Legehennen	ist	heftiger	politischer	Streit	entbra
lemma	um	der	deutsch	Legehenne	sein	heftig	politisch	Streit	entbre
POS	AP	ART	ADJA	NN	VAF	ADJA	ADJA	NN	VVPP
cs 3	S								
cs 2	NP(1)				NP(2)				
cs 1	PP								

(2)

[1]	1	2	3	4	5	6	7	8	9
words	Um	die	deutschen	Legehennen	ist	heftiger	politischer	Streit	entbra
lemma	um	der	deutsch	Legehenne	sein	heftig	politisch	Streit	entbrer
POS	AP	ART	ADJA	NN	VAF	ADJA	ADJA	NN	VVPP

Interaktionen

Es werden folgende Interaktionen für sinnvoll gehalten:

- (1) Zeilen/Zeilengruppen ein-/ausblenden
- (2) Zeilen highlighten
- (3) Referenzzeile hinzufügen/löschen
- (4) Leerzeile hinzufügen/löschen
- (5) Leerzellen anzeigen/ausblenden
- (6) Zeile verschieben
- (7) Gruppierung zu Ebenensets
- (8) Label in Kurzform oder in Langform
- (9) Daten in Kurzform oder in Langform
- (10) Informationen zum Annotationsset einblenden

- (11) Spalten highlighten
- (12) einzelne Zellen highlighten
- (13) zusammengehörige Elemente highlighten
- (14) Segmentierung ändern
- (15) Zeilenumbruchoption wählen (kein Umbruch / pro Satz eine Zeile)
- (16) Positionsvergleich (bei Satzzeilenumbruch): Text ausblenden + Spalten mit fester gleicher Breite

Problem: Leerzeilen ein/ausblenden

Problem : Was passiert, wenn Leerzeilen ausgeblendet wurden, daraufhin die Reihenfolge verändert wurde und nun die Leerzeilen wieder eingeblendet werden? Je nach Umfang der Reihenfolgen-Änderungen kann sich ein mehr oder weniger unerwartetes (chaotisches) Bild ergeben; ein Wiedereinblenden wäre also nur sinnvoll, wenn sich die Reihenfolge in der Zwischenzeit nicht verändert hat. Für den Fall, dass sich nun aber die Reihenfolge nach einem Ausblenden verändert hat, gibt es zwei Möglichkeiten, den Nutzer auf dieses Problem vorzubereiten: (1) er erhält gar nicht erst die Möglichkeit, die Zeilen wieder einzublenden, z.B. wird der Link zum Einblenden deaktiviert (und ausgegraut); Nachteil: der Nutzer weiß nicht, warum der Link deaktiviert ist; (2) der Nutzer erhält nach Klick auf den Link zum Wiedereinblenden eine Nachricht angezeigt, mit dem Hinweis, dass das Einblenden der Leerzeilen möglicherweise nicht den gewünschten Effekt hat; er kann wählen zwischen der Möglichkeit, die Leerzeilen trotzdem wieder einzublenden, und der Möglichkeit, die (unsichtbaren) Leerzeilen zu löschen und stattdessen neue Leerzeilen zu erstellen (eine Abbruch-Möglichkeit sollte natürlich ebenfalls vorhanden sein).

1 Visualisierung von hierarchischen Annotationen (Syntax)

Da hierarchische Daten am effektivsten über Node-Link-Diagramme dargestellt werden können (vgl. Ware 2004 :210ff.), sollten die hierarchischen Annotationen in dieser Form dem Nutzer zur Verfügung gestellt werden. Aufgrund des Platzbedarfes von Graphen ist davon auszugehen, dass die Nutzer von der Möglichkeit profitieren würden, wenn sie die Anzeigart der Darstellung (Ebenen in der intralinen Darstellung vs. Graph)⁴² selbst bestimmen könnten.

Problem der kreuzenden Kanten: Die Visualisierung hierarchischer Daten in einem Browser unterliegt technischen Beschränkungen: Kreuzende Kanten sind mit XHTML nicht darstellbar. Mit SVG wäre diese Art der Darstellung dagegen möglich (wie TIGER zeigt) – und man denkt unweigerlich an die vielen Nutzungsmöglichkeiten, die diese Technik im Kontext mit anderen Visualisierungen mit sich bringen könnte. Tatsächlich sind die aktuellen Browser in der Lage, SVG-Dateien darzustellen (der IE7 über ein Plugin, Mozilla mit nativer Unterstützung); eine Interaktion ist jedoch gebunden an ein Zusammenspiel von SVG, CSS, DOM/Javascript – und hier sind die Browser bisher noch nicht einsatzfähig.

Es stehen zwei Möglichkeiten zur Auswahl, möchte man die syntaktische Struktur dennoch visualisieren: (1) Man macht Gebrauch von der Möglichkeit, die originalen (aber nicht interaktiven) SVG-TIGER-Grafiken in die XHTML-Datei einzubinden; (2) man verzichtet auf die Darstellung von kreuzenden Kanten und erstellt Graphen mit den Mitteln, die in XHTML zur Verfügung stehen. Ersteres ist sinnvoll, wenn die detaillierte Darstellung des Baumes eine übergeordnete Rolle spielt; letzteres dagegen, wenn eine Annotation interaktiv hervorgehoben werden soll und die Positionierung dieses Wertes innerhalb der syntaktischen Struktur überprüft werden soll.

Eine SVG-Grafik in eine XHTML-Datei einzubinden ist nicht weiter problematisch, da es sich bei beiden Datentypen um XML-Derivate handelt. Einen Graphen innerhalb einer intralinen Darstellung in XHTML zu erzeugen, ist dagegen nicht ohne „Tricks“ möglich, da es für diese Art der Präsentation in XHTML keine definierten Elemente gibt.

Eine Lösung wird im Folgenden skizziert (sie nutzt XHTML und CSS).

⁴² Eine intralinea Visualisierung der Struktur über indizierte Klammerung ist ebenfalls denkbar, wird aber wegen des negativen Effektes auf die Lesbarkeit als zweitrangig angesehen.

Umsetzung

Struktur

```
<tbody class="tree top">
  <tr class="data">
    <th>cs 3</th>
    <td class="tools"></td>
    <td class="first_c last_c" colspan="10"><span>S</span></td>
  </tr>
  <tr class="line">
    <td class="row_label">&nbsp;</td>
    <td></td>
    <td class="first_c line_right" colspan="4">
      <!--Linien Anfang-->
      <div><div class="vline"><div>&nbsp;</div></div>
      <div class="hline">&nbsp;</div></div>
      <!--Linien Ende--></td>
    <td class="line_center">
      <!--Linien Anfang-->
      <div><div class="vline"><div>&nbsp;</div></div>
      <div class="hline">&nbsp;</div></div>
      <!--Linien Ende--></td>
    <td class="line_center" colspan="3">
      <!--Linien Anfang-->
      <div><div class="vline"><div>&nbsp;</div></div>
      <div class="hline">&nbsp;</div></div>
      <!--Linien Ende--></td>
    <td class="line_center">
      <!--Linien Anfang-->
      <div><div class="vline"><div>&nbsp;</div></div>
      <div class="hline">&nbsp;</div></div>
      <!--Linien Ende--></td>
    <td class="last_c line_left">
      <!--Linien Anfang-->
      <div><div class="vline"><div>&nbsp;</div></div>
      <div class="hline">&nbsp;</div></div>
      <!--Linien Ende--></td>
  </tr>
  <tr class="data">
    <td class="row_label">cs 2</td>
    <td></td>
    <td class="first_c" colspan="4"><span>NP(1)</span></td>
    <td><div class="vline"><div>&nbsp;</div></div></td>
    <td colspan="3"><span>NP(2)</span></td>
    <td><div class="vline"><div>&nbsp;</div></div></td>
    <td class="last_c"><div class="vline"><div>&nbsp;</div></div></td>
  </tr>
</tbody>
```

XHTML-Code (Ausschnitt): Zeilengruppe mit fester Zeilenabfolge

Alle Ebenen der Gruppe „Konstituente“ befinden sich in einer gemeinsamen Zeilengruppe (**tbody**), die mit der Klasse **class="tree"** gekennzeichnet ist. Zeilen in dieser Zeilengruppe befinden sich in einer festgelegten Abfolge. Je nachdem, ob sich die Zeile mit dem Wurzelknoten an erster oder an letzter Zeilenposition befindet (also oben oder unten angezeigt wird), enthält die Zeilengruppe zusätzlich die Klasse „**top**“ bzw. „**bottom**“. Diese Angabe

ist notwendig, damit die horizontalen Linien entweder oben oder unten in ihrer Zelle ausgerichtet werden. Der Großteil der Verbindungslinien (Kanten) ist in den Zeilen mit der Klasse **class="line"** lokalisiert, die sich zwischen den Datenzeilen (**class="data"**) befinden. Zellen der Datenzeile, die keine Daten beinhalten, sind ebenfalls mit einer (vertikalen) Linie gefüllt.

Um die Linien zu erzeugen hätte es verschiedenen Möglichkeiten gegeben: Die Linien hätten als Bilder `` mit variabler (CSS)Breitenangabe oder als Hintergrundbilder in die Zellen eingefügt werden können (ebenfalls über CSS). Beide Varianten funktionieren und sind auf struktureller Ebene einfacher als die gewählte Variante. Die gewählte Variante hat allerdings den entscheidenden Vorteil, dass die Farbe, Stärke und Art der Linien über eine CSS-Angabe geändert werden können. Dies ist wichtig, wenn z.B. einzelne oder zusammengehörige Teilbäume hervorgehoben werden sollen.

Horizontale und vertikale Linien sind kodiert über `<div>`-Tags (`<div class="vline">` `<div> </div>` `</div>` bzw. `<div class="hline">` ` </div>`)⁴³. Die visuelle Linien-Eigenschaft erhalten sie erst über die Formatierung in CSS. Für die Darstellung der horizontalen Linien werden dabei drei Versionen benötigt (a) die Linie ist links ausgerichtet und hat eine Breite von 50% (**class="line_left"**) (b) die Linie ist mittig ausgerichtet und hat eine Breite von 100% (**class="line_center"**) (c) die Linie ist rechts ausgerichtet und hat eine Breite von 50% (**class="line_right"**). Diese Angaben sind in CSS kodiert.

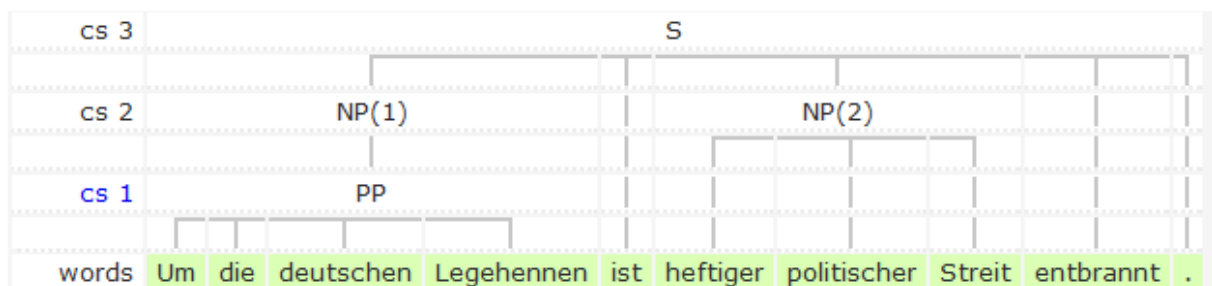


Abbildung 12: interlineare Tabelle mit XHTML und CSS

Interaktion

Für die intralinere Baumdarstellung sind folgende Interaktionen sinnvoll:

- (1) Wechsel von der Ebenendarstellung in die Baumdarstellung et vice versa (mit oder ohne Neuladen der Seite)

⁴³ Die Verschachtelung bei der vertikalen Linie ist notwendig, um in Firefox die Überlagerung der vertikalen und horizontalen Linien zu ermöglichen!

- (2) Spiegelung der Hierarchieebenen in der Vertikalen (= Wurzelknoten oben vs. Wurzelknoten unten)
- (3) Vergleich von Bäumen innerhalb eines Datensatzes (z.B. IP und Syntax)
- (4) Hervorheben von zusammengehörigen Segmenten (bei gesplitteten Elementen)
- (5) Hervorheben eines Teilbaums: (a) ein einziger Teilbaum (b) zusammengehörige Teilbäume bei gesplitteten Elementen
- (6) Auswahl eines Teilbaums: (a) ein einziger Teilbaum (b) zusammengehörige Teilbäume
- (7) Ein/Ausblenden von Teilbäumen: (a) ein einziger Teilbaum (b) zusammengehörige Teilbäume bei gesplitteten Elementen
- (8) komprimierte Darstellung vs. unkomprimierte Darstellung: = zwei Bäume gleicher Struktur (Segmentierung + Hierarchiestufen) ineinander verschränken (z.B. Konstituenten und Satzfunktion)
- (9) Drucken/Export: ganzer Baum/Teilbäume; verschiedene Vergrößerungsstufen ohne Qualitätsverlust

Erweiterungsmöglichkeiten

Erweiternd wäre eine Überlagerung mit SVG-Linien denkbar, die nach Auslesen der Annotationspositionen (per DOM/JavaScript) absolut positioniert auf einem höheren Layer (z-index) positioniert werden. Damit wäre die Visualisierung von Spuren möglich. Bei einer Erweiterung mit SVG sollte bedacht werden, dass nicht jeder potenzielle Nutzer das Adobe SVG-Plugin installiert bzw. einen SVG-fähigen Browser hat.

Ein Problem für die interlineare Baumdarstellung ergibt sich bei der umgebrochenen Partiturvariante: Ein umgebrochener Baum ist vermutlich nicht gut lesbar. Hier ist es also angeraten, per Default-Einstellung eine Zeilenschreibweise anzuzeigen.

2 Interlineare Darstellung von Position in Ergebnislisten

Um, wie in Projekt B4 gefordert, die Exploration von Wortstellungsmustern zu ermöglichen, wird eine interlineare Darstellung innerhalb einer Ergebnisliste vorgeschlagen. In der Ergebnisliste ist ein direkter Bezug zur Position gegeben, sowie die Möglichkeit, die Abfolge von Annotationen über eine farbliche Markierung hervorzuheben.

	1	2	3	4	5	6	7	8	9	10	11	12[token]
1 <input type="checkbox"/> T_035_29	uuarun	thō	hirta	In	thero	landskeffi	.					
	vfin	temp	new	su	loc							
2 <input type="checkbox"/> T_036_11-12	thō	sliumouuard	thar	mit	themo	engile/	menigihimilischesheres.					
			vfin		giv		new					
			su				su					
3 <input type="checkbox"/> T_037_23	senonutho	uuas	man	In	hierusalem							
			new									
			vfin	su	predn							
4 <input type="checkbox"/> T_38_22	uuas	thō	thār	anna	uuizzaga							
				new								
	vfin	temp		su								
5 <input type="checkbox"/> T_39_17	thō	thaz	gihorta	herodesther	cuning							
		giv		new								
		do	vfin	su								
6 <input type="checkbox"/> T_51_16-17	thero	uuas	andreas	bruoder	/	simonis	p&ri					
			new									
	io	vfin	su									
	1	2	3	4	5	6	7	8	9	10	11	12[token]

Diese Darstellung wäre noch sinnvoller, wenn es die Möglichkeit gäbe, die Liste nach Typen zu sortieren, bzw. als ersten Schritt auf eine Suchanfrage nur die Typen präsentiert zu bekommen mit der Option, sich in einem zweiten Schritt die einzelnen Vertreter eines Typs anzusehen. Dies ist zwar eine Möglichkeit, die von der befragten Nutzerin in einem Cognitive Walkthrough als sehr nutzerfreundlich bewertet wurde, allerdings ist eine Realisierung von der Abfragesprache her derzeit nicht machbar.

6 Basisvisualisierung: extralineare Darstellung (Baum)

1 Extralineare Darstellung von Bäumen: RSTTool (horizontal)

RST-Visualisierungen, die mit Hilfe des RSTTools erstellt wurden, können exportiert und in webfähige Formate (z.B. jpg) transformiert werden. Die Dateien können als Bilddaten in die Nutzeroberfläche integriert werden. Sie haben folgende Nachteile: (1) Es können keine Daten in diesen Grafiken hervorgehoben werden (sei es als Ergebnis auf eine Suchanfrage hin oder interaktiv, um z.B. eine neue Suchanfrage zu formulieren). (2) Die Grafiken sind unter Umständen sehr raumgreifend, können aber nicht mehr wie im RSTTool in Teilbereichen zusammengeklappt werden. (3) Sie sind nicht interaktiv und damit nicht als Navigation für längere Texte eingesetzt werden.

2 Extralineare Darstellung von Bäumen: RST vertikal

Im Gegensatz zur RSTTool-Darstellung wird bei der XHTML-basierten vertikalen Variante keine grafische Unterscheidung getroffen zwischen parataktischen und hypotaktischen Beziehungen (Abbildung 13). Stattdessen werden alle Beziehungen gleich (eckig) dargestellt.

0	1	2	3	4	5	s	RST-Segment
						1	So träge wie allgemein oft angenommen ,
						1	scheint die Jugend im Fläming nun doch nicht zu sein .
						2	Denn wenn sich jetzt unterstützt von Schülern auch in Wiesenburg ein Förderverein für die Grund- und Gesamtschule gründet , zeigt dies ,
						2	welchen Stellenwert einer guten schulischen Ausbildung inzwischen eingeräumt wird .
						3	Wie anderswo bereits seit längerem erfolgreich praktiziert , wird die Schule so stärker in den Mittelpunkt der Gesellschaft gerückt .
						4	Sie ist kein abgeschottetes Gebilde mehr , sondern Teil der Kommunen .
						5	An die Stelle von "die da oben müssten mal ... " ist Eigeninitiative getreten .
						6	Selbst etwas anpacken und nicht auf Allmosen warten , scheint nun die Devise zu sein .
						7	Das ist gut so .
						8	Bleibt zu hoffen , dass sich nicht nur dem Wiesener Förderverein möglichst viele Menschen aus diversen Gruppen der Gesellschaft anschließen .
						9	Beispielsweise könnten Handwerker und Betriebe der Region so frühzeitig Kontakte zu künftigen Lehrlingen knüpfen oder im Gespräch mit Lehrern ihre Anforderungen an die oft kritisierte Ausbildung in den Schulen verdeutlichen .
						10	Dass für eigentlich normale Dinge im Schulalltag dennoch immer auch Spendengeld nötig bleibt , ist ein Wermutstropfen

Abbildung 13: Extralineare Darstellung von RST

Dies liegt in der Tatsache begründet, dass XHTML und CSS keine Möglichkeiten

bereitstellen, Bögen zu zeichnen. Hier wäre SVG von Vorteil. Wie bereits angemerkt kann diese Technik aber nicht vorausgesetzt werden. Auf die gewohnte Darstellung einer hypotaktischen Beziehung per Bogensegment mit Pfeil wurde also verzichtet. Richtungen über Pfeile zu visualisieren wäre zwar in XHTML (über Umwege) realisierbar, jedoch hätte ein Einfügen von (wahrnehmbaren) Pfeilen zur Folge, dass die recht kompakte Darstellung ausgeweitet würde und so der visuelle Bezug zu den Textsegmenten (abhängig von der Hierarchietiefe) verloren ginge. Der entstandene Informationsverlust kann möglicherweise dadurch ausgeglichen werden, dass bei Mouseover über einem Knoten die zugehörigen Textsegmente unterschiedlich (bei subordinierenden Textsegmenten) oder gleich (bei koordinierenden Textsegmenten) markiert werden.

Zugunsten der Kompaktheit werden auch die textuellen Annotationswerte (z.B. „elaboration“ oder „coordination“) nicht angezeigt. Diese Informationen sollten aber bei Bedarf (z.B. bei Mouseover über einem Knoten) verfügbar gemacht werden. Außerdem sollte der Nutzer die Möglichkeit erhalten, die Anzeige der textuellen RST-Annotationen dauerhaft ein- oder auszublenden.

Dieser Darstellungstyp ist nicht auf den Bereich der Diskursstruktur beschränkt. Er ließe sich genauso gut auf andere Bäume (z.B. Syntax) anwenden.

7 Basisvisualisierung: superlineare Darstellung

Als letzte Basisvisualisierung soll im Folgenden eine superlineare Darstellung konzipiert werden, die – wie unten dargestellt – eng mit einer intralinen Darstellung verknüpft ist. Die superlineare Darstellung wird aktiviert, indem ein Token des Primärtextes mit der Maus überfahren wird. Als Folge werden die Annotationen zu diesem Token in einem Bereich eingeblendet, der direkt am Aktionsort positioniert ist (ähnlich einem < title>-Attribut) (s. Abbildung 14).

Glückliche Hühner Um die deutschen Legehennen ist heftiger politischer Streit entbrannt . Bundesagrarministerin Renate Künast will das Halten der Tiere in engen Legebatterien bereits vom Jahr 2006 an verbieten . In den EU-Nachbarländern soll das erst fünf Jahre später gelten . Eier-Produzenten aus der ganzen Republik machen gegen Künasts Pläne mobil . Die Betriebe im Osten fürchten , dass die hohen Investitionen , die sie in moderne europäischem Standard gesteckt haben , umsonst gewesen sind . vor , ausländische Konkurrenten könnten dann mit Billig-Eiern aus Markt überschwemmen . Beide Bedenken sind nicht einfach von der Künast muss zumindest über längere Übergangsfristen nachdenken Käfighaltung abzuschaffen , sollte sie nicht rütteln . Artgerechte Tierhaltung zu den Eckpunkten der von ihr propagierten Agrarwende . Und schließlich greifen immer mehr Verbraucher beim Einkauf bewusst zu den tierfreundlich erzeugten Eiern .

Sent: s
Focus_newInf: nf-unsol
PP: PP
Inf-Stat: giv-inactive
NP: NP
Click to highlight

Abbildung 14: superlineare Darstellung

Da es – bedingt durch die verschiedenen Annotationsebenen – mehrere Segmentierungsraster gibt, die auf ein und denselben Text projiziert werden, kann der Computer zunächst einmal nicht wissen, welches Segment bei Überfahren eines Token gemeint ist (es sei denn, die Segmentierung wurde vorher festgelegt). Infolgedessen kann es vorkommen, dass das Token sich hinsichtlich eines Annotationssets in einem Segment mit mehreren Token befindet, ohne dass dessen Ausdehnung automatisch angezeigt werden könnte. Um diesen Informationsverlust auszugleichen, wird dem Nutzer die Möglichkeit gegeben, sich anzeigen zu lassen, auf welches Textsegment sich eine der eingeblendeten Annotationen bezieht:

Wenn der Nutzer im eingeblendeten Bereich eine Annotation auswählt (per Klick oder per Mouseover), wird das dazugehörige Textsegment hervorgehoben (je nach Aktionstyp dauerhaft oder temporär).⁴⁴ Über eine Zusatzoption kann sich der Nutzer alle Textsegmente anzeigen lassen, die hinsichtlich der gewählten Annotationsebene annotiert wurden (z.B. alle Segmente, die auf der Ebene „givenness“ annotiert sind). Weiterhin kann er sich

⁴⁴Die superlineare Darstellung ist damit notwendigerweise eng gekoppelt an eine intralinea Darstellung.

alle Textsegmente anzeigen lassen, die hinsichtlich der Annotationsebene mit dem gleichen Wert annotiert wurden (z.B. alle neuen Diskursreferenten, beschrieben über „new“). Da während der Datenexploration oftmals verschiedene Annotationsebenen zueinander in Beziehung gesetzt werden sollen, ist es wie bei der intralinearen Darstellung von Vorteil, wenn der Nutzer den Stil der Hervorhebung selbst bestimmen kann. Dies soll möglich sein über eine Erweiterung des eingeblendeten Feldes (s.u.).

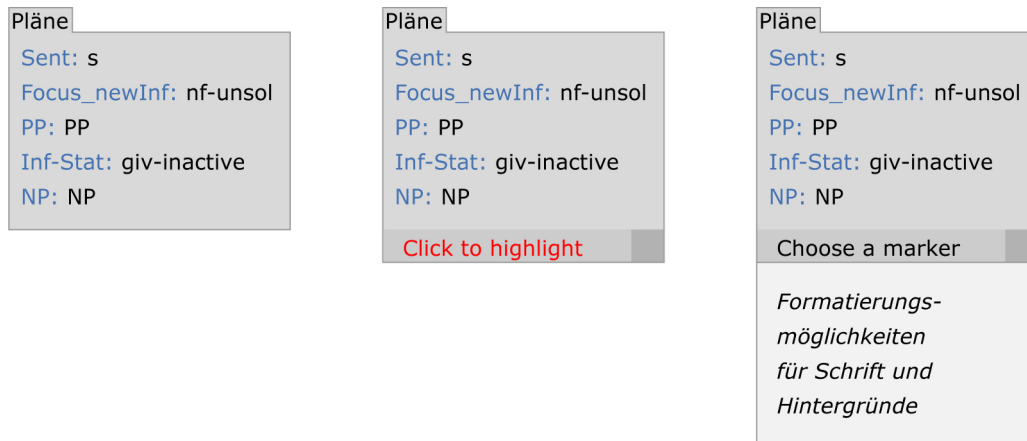


Abbildung 15: superlineare Darstellung: Interaktionsphasen

Diese Visualisierung macht extensiven Gebrauch von der Interaktionstechnik der direkten Manipulation. Diese wird von Nutzern in der Regel als positiv bewertet (vgl. Shneiderman & Plaisant 2005, Ware 2004). Ob das vorgestellte Konzept effizient bei der Exploration von Annotationen ist, müsste in Nutzertests überprüft werden.

8 Komplexe Darstellung reich annotierter linguistischer Daten

Eine komplexe Darstellung ermöglicht die gleichzeitige⁴⁵ Exploration heterogener Rohdaten und ihrer heterogenen Annotationen. Es können RST, Syntax, Koreferenz, einfache Attribut-Wert-Paare und Position gleichzeitig dargestellt und untersucht werden. Dazu werden Visualisierungstypen auf einem Screen (1) miteinander kombiniert und/oder (2) ineinander verschränkt (=integriert). Bei der Kombination von Visualisierungstypen werden mehrere Visualisierungsbereiche auf einem Screen dargestellt, z.B. ein Bereich mit extralinearer Darstellung für Syntaxdaten, ein Bereich mit superlinearer Darstellung für Koreferenzdaten, ein Bereich mit interlinearer Darstellung für sonstige Attribut-Wert-Paare. Bei der integrativen Darstellung dagegen sind die Visualisierungsformen miteinander verschränkt, z.B. RST/Koreferenz oder Partitur/Syntax (s.u.). Die kombinierten Visualisierungsbereiche sind *koordiniert*, in dem Sinne, dass eine Aktion in einem Bereich Auswirkungen hat auf die dargestellten Inhalte oder Darstellungsform im anderen Bereich.

Entwurf für eine kombinierte und integrative Darstellung

Der Bereich, der für die Detailansicht zur Verfügung steht, wird wie bei der kombinierten Darstellung auf mehrere Anzeigebereiche aufgeteilt; diese Anzeigebereiche können einfache Visualisierungstypen beinhalten (z.B. vertikaler Baum) oder aber integrative Darstellungen (z.B. vertikaler Baum+Koreferenz).

Eine für den Nutzer vorteilhafte Aufteilung ist möglichst variabel und skalierbar. Die Variabilität kann über eine flexible Belegung der Anzeigebereiche gewährt werden; die Skalierbarkeit der einzelnen Bereiche unterliegt hingegen technischen Beschränkungen:

Eine Technik, die es erlaubt, dass die einzelnen Bereiche vom Nutzer verkleinert oder vergrößert werden können, ist die Frametechnik. Frames sind in XHTML 1.1 allerdings nicht mehr erlaubt. Da eine integrierte Anzeige von SVG-Grafiken (TIGER) als sinnvoll erachtet wird, und darüber hinaus eine Integration zukünftiger interaktive SVG-Grafiken nicht ausgeschlossen werden soll, wird XHTML jedoch der HTML-Frame-Variante vorgezogen. Möglicherweise bietet auch die Technik XUL Funktionalitäten zur Skalierung. Dies wurde hier nicht überprüft.

Alternativ kann folgendes Konzept zur Anwendung kommen: Der Bereich des Dokuments wird aufgeteilt in einen linken und einen rechten Containerbereich (Abbildung 16). Der linke Bereich ist in der Vertikalen orientiert, der rechte Bereich je nach verbleibendem

⁴⁵ Denkbar und u.U. nützlich wäre auch eine sequentielle Abfolge z.B. über Karteireiter. Von dieser Art der Kombination bin ich bereits für die Darstellung von Übersicht, Ergebnisliste und Detailansicht ausgegangen.

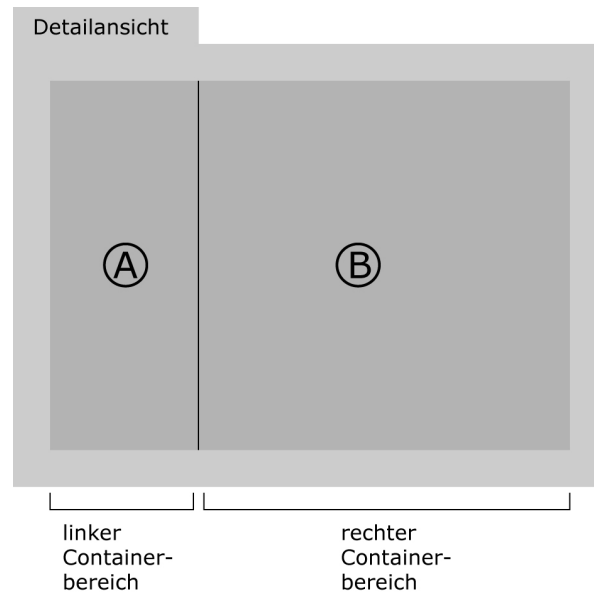


Abbildung 16: linker und rechter Containerbereich

Platz eher in der Horizontalen. Damit wird eine gleichzeitige Ansicht von vertikalen Bäumen (z.B. RST) und horizontaler Partitur (mit Annotationen zur Informationsstruktur) möglich. Sinnvollerweise lässt sich das Verhältnis der Spaltenbreiten umkehren, so dass z.B. Audio/Videodaten in einem schmaleren Bereich auf der rechten Seite angezeigt werden können und der zentrale mittlere Bereich mit den Annotationen gefüllt ist, deren Fokussierung unterstützt werden soll. Wenn der linke Bereich ausgeblendet wird, nimmt der rechte Bereich automatisch die gesamte Breite ein; wenn der rechte Bereich ausgeblendet wird, nimmt der linke Bereich automatisch die gesamte Breite ein.

Der rechte und der linke **Containerbereich** können jeweils mehrere **Visualisierungsbereiche** enthalten (Abbildung 17). Die Anzahl der Visualisierungsbereiche pro Container kann zwischen 0 und n variieren.

Für jeden Visualisierungsbereich gilt:

- Er kann einen beliebigen Inhalt zugewiesen bekommen (nicht nur die Basisvisualisierungen, sondern auch ein Bild oder einen Video-Abspielbereich und beliebige andere Inhalte; dies gewährleistet auch zukünftig eine Erweiterbarkeit um neue Module)
- Er kann ausgeblendet werden (durch Klick auf eine Schaltfläche in der rechten oberen Ecke des Bereiches). Ausgeblendete Visualisierungsbereiche bleiben an ihrer Position, werden jedoch in der Vertikalen zu einer einfachen Zeile minimiert.
- Als reaktivierbare Zeilen bleiben sie in der Ansicht vorhanden. (Die Reaktivierung erfolgt z.B. durch Klick auf die Zeile)

- Er kann eingeblendet werden (über eine Auswahl im Menü „Darstellung ändern“).
- Er kann gelöscht werden (durch Klick auf einen „Schließen“-Button in der rechten oberen Ecke des Bereiches)
- Er kann in einen anderen Container bzw. innerhalb eines Containers von oben nach unten et vice versa verschoben werden (nach Aktivierung von „Darstellung ändern“).
- Er kann in seiner Ausprägung variiert werden (nach Aktivierung von „Darstellung ändern“).

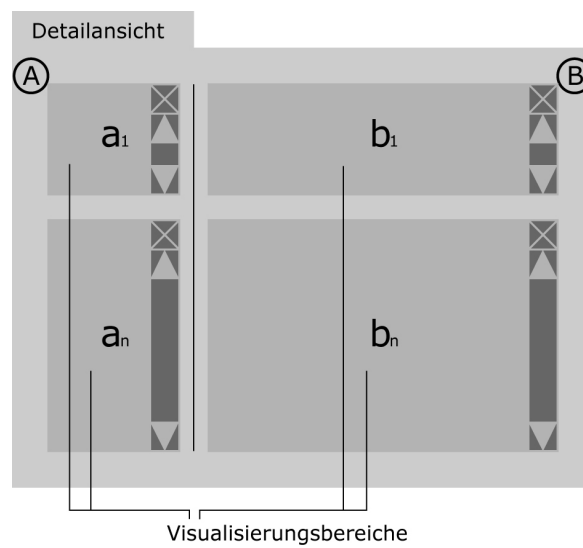
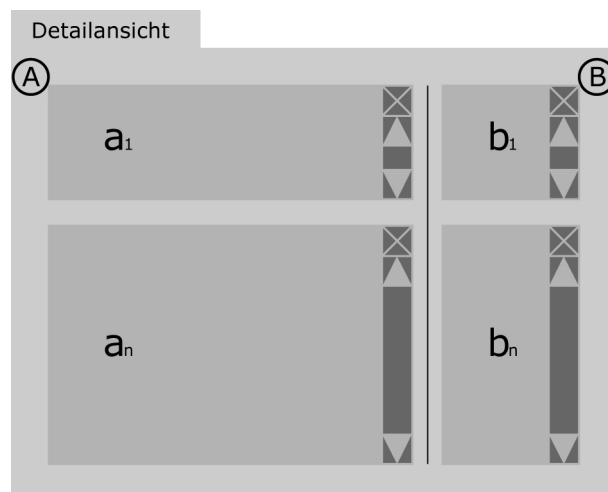
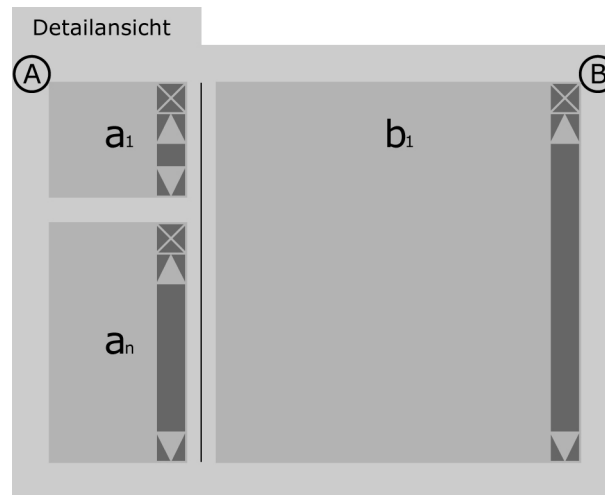


Abbildung 17: Visualisierungsbereiche (Varianten)





Nach der Aktivierung der Schaltfläche „Darstellung ändern“ erscheinen auf der Oberfläche zusätzliche (eher auffällige) Schaltflächen, mittels derer die Konfiguration der bestehenden Darstellung geändert werden kann. Als Schaltflächen können z.B. einfache Pfeil-Buttons in der rechten oberen Ecke jedes Visualisierungsbereichs eingesetzt werden, mittels derer der Bereich (a) innerhalb eines Containerbereichs nach oben oder unten oder (b) von einem Container in den anderen verschoben werden kann. Weiterhin sollte die Möglichkeit gegeben sein, neue Visualisierungsbereiche hinzuzufügen und diese zu konfigurieren. Das Löschen eines Visualisierungsbereichs erfolgt über eine Schaltfläche in der rechten oberen Ecke eines Visualisierungsbereiches.

Da vermutlich aus Nutzersicht die Begriffe extralinear, intralinear und superlinear nicht auf den ersten Blick eingängig sind, soll hier der Kreis geschlossen werden zurück zu den prototypischen Visualisierungsformen aus den bekannten Tools; diese Prototypen können sich aus folgenden Visualisierungstypen zusammensetzen:

Typ 1: vertikaler Baum (extralinear, intralinear und superlineare Darstellung – evtl. erweiterbar auf interlineare Darstellung)

Typ 2: Partitur (interlineare und intralinear Darstellung) – in der interlinearen Darstellungen können möglicherweise Zeilen zusammengemappt werden, beispielsweise könnten bei der Annotationsebene Informationsstruktur der Hintergrund über Hintergrundfarbe und Fokussierung über Fettung oder grafisches Symbol angezeigt werden

Typ 3: Textblock (intralinear und superlineare Darstellung)

Typ 4: horizontaler Baum (interlineare Darstellung oder Bild)

Beispiel für eine integrierte Darstellung (2fach-Kombinationen)

Folgende Visualisierung integriert die intralineare Darstellung von Koreferenz in einen extralinen vertikalen Baum (RST).

0	1	2	3	4	5	RST-Segment
					1	So träge wie allgemein oft angenommen ,
					1	scheint die Jugend im Fläming nun doch nicht zu sein .
					2	Denn wenn sich jetzt unterstützt von Schülern auch in Wiesenburg ein Förderverein für die Grund- und Gesamtschule gründet , zeigt dies ,
					2	welchen Stellenwert einer guten schulischen Ausbildung inzwischen eingeräumt wird .
					3	Wie anderswo bereits seit längerem erfolgreich praktiziert , wird die Schule so stärker in den Mittelpunkt der Gesellschaft gerückt .
					4	Sie ist kein abgeschottetes Gebilde mehr , sondern Teil der Kommunen .
					5	An die Stelle von "die da oben müssten mal ... " ist Eigeninitiative getreten .
					6	Selbst etwas anpacken und nicht auf Allmosen warten , scheint nun die Devise zu sein .
					7	Das ist gut so .
					8	Bleibt zu hoffen , dass sich nicht nur dem Wiesener Förderverein möglichst viele Menschen aus diversen Gruppen der Gesellschaft anschließen .
					9	Beispielsweise könnten Handwerker und Betriebe der Region so frühzeitig Kontakte zu künftigen Lehrlingen knüpfen oder im Gespräch mit Lehrern ihre Anforderungen an die oft kritisierte Ausbildung in den Schulen verdeutlichen .
					10	Dass für eigentlich normale Dinge im Schulalltag dennoch immer auch Spendengeld nötig bleibt , ist ein Wermutstropfen

Konzept: technische Realisierung

Es stehen diverse Möglichkeiten zur Verfügung, die Original-XML-Daten (+ Bild, + Video, + Audio) im Browser darzustellen. Die naheliegendste Variante wäre es, die XML-Daten direkt an den Browser auszuliefern und von diesem interpretieren zu lassen. Da es sich bei den XML-Daten um eine rein semantische Auszeichnung ohne Formatierungsanweisung handelt, würde dies allerdings bedeuten, dass der Browser „raten“ müsste, wie die Daten dargestellt werden sollen. Statt zu raten, zeigen die Browser (FF 2.0, IE 7) einfach den XML-Baum an⁴⁶. Dem Browser können jedoch Formatierungsanweisungen mitgeliefert werden, die eine formatierte Darstellung der XML-Daten im Browser ermöglichen. Zwei Methoden stehen zur Verfügung, die sich in ihrer Formatierungswirkung ergänzen, aber auch einzeln angewandt ein visuelles Ergebnis im Browser erzeugen: dies sind XSLT und CSS. Dabei kann XSLT die XML-Daten u.a. in die gewünschte Reihenfolge bringen, CSS kann die Daten u.a. visuell auszeichnen. Voraussetzung ist, dass die Browser in der Lage sind, XSLT und CSS zu interpretieren. Hier befinden sich die Schwachstellen: Damit ein XSL-Stylesheet vom Browser interpretiert werden kann, muss er über einen XSLT-Prozessor verfügen, der die XML-Daten und die XSL-Daten zueinander in Beziehung setzen kann. Die derzeit integrierten Prozessoren (wie z.B. FF 2.0 und IE 7) sind jedoch recht langsam und unterstützen noch nicht die neuesten Standards (XSLT 2.0, XPath 2.0). Diese neuen Standards beinhalten wesentliche Erweiterungen gegenüber den ersten Versionen und wären sinnvoll bei großen Datenmengen und Datensets (z.B. die neue Funktion „for-each-group“). Die Browserunterstützung von Cascading Stylesheets im Zusammenhang mit XML ist noch nicht browserübergreifend gut.

XML hätte gegenüber XHTML den großen Vorteil, dass die Vielzahl der Annotationen im Quellcode erhalten bleibt, ohne dass diese erst in XHTML-kompatible Ersatzrepräsentationen umgewandelt werden müssten (z.B. in Attributwerte von title, class, id). Damit die Annotationen vom Nutzer gezielt markiert (=formatiert) werden können, muss allerdings gewährleistet sein, dass (a) das Zusammenspiel von XML und CSS funktioniert, (b) das Zusammenspiel von XML und DOM/JavaScript funktioniert und (c) das Zusammenspiel von XML, DOM/JavaScript und CSS funktioniert (was nicht automatisch selbstverständlich ist).

⁴⁶ Opera zeigt hingegen alle Textstrings als Text an, erzeugt also einen Fließtext, aber ohne weitere Formatierung.

Die Visualisierung reich annotierter linguistischer Daten wäre sehr viel eleganter mit XML als mit XHTML zu lösen. Da jedoch die üblichen Browser in den heutigen Versionen die Darstellung von formatiertem XML noch nicht ausreichend unterstützen, wird die Realisierung mittels XHTML vorgenommen. Sollten in späteren Versionen die Browser die XML-Daten besser darstellen können, so wäre es angeraten, dann auf die Funktionalitäten von XML zurückzugreifen.

Unterdessen kann eine Realisierung mittels XHTML erfolgen, was gegenüber reinem HTML den Vorteil mit sich bringt, SVG (und langfristig SMIL?) und andere Dateninseln (auch XML-Inseln) direkt in die Datei einbetten zu können.

Modularisierung

Im Sinne der Wiederverwendbarkeit und Wartbarkeit wird eine modulare Realisierung in einzelnen Komponenten angestrebt. Pro Visualisierungstyp soll es eine Komponente geben. Da ein Visualisierungstyp mit anderen Visualisierungstypen verschränkt werden kann, muss definiert werden, welche Komponente im Fall einer Verschränkung die determinierende ist, sprich: welcher Visualisierungstyp im Fall einer Verschränkung die basale XHTML-Struktur (inkl. CSS-Stilen und angelegten Interaktionsmöglichkeiten) liefert, in die die andere Komponente eingebettet werden kann.

Folgende Komponenten sind möglich:

1. intralinear -> XHTML: Textblock
2. interlinear -> XHTML: Tabelle
3. extralinear -> XHTML: Tabelle
4. superlinear -> XHTML: <title>-Attribut oder ein on-the-fly über Javascript/
DOM erzeugter positionierter Bereich

Folgende Komponentenkombinationen sind möglich:

1. intralinear + interlinear -> interlinear ist dominant (Die grundlegende XHTML-Struktur wird bestimmt durch die Tabelle)
2. intralinear + extralinear -> extralinear ist dominant (Die grundlegende XHTML-Struktur wird bestimmt durch die Tabelle)
3. intralinear + superlinear -> intralinear ist dominant (Die grundlegende XHTML-Struktur wird bestimmt durch den Text)
4. interlinear + extralinear -> extralinear ist dominant (Die grundlegende XHTML-

Struktur wird bestimmt durch die Extralinear-Tabelle)

5. interlinear + superlinear -> interlinear ist dominant (Die grundlegende XHTML-Struktur wird bestimmt durch die Tabelle)
6. extralinear und superlinear -> extralinear ist dominant (Die grundlegende XHTML-Struktur wird bestimmt durch die Tabelle.

Jede dieser Visualisierungskomponenten bringt ihre spezifischen Möglichkeiten der Interaktion mit sich. Zum Beispiel können in der interlinearen Darstellung Zeilen verschoben werden und in der extralinen Darstellung (z.B. RST) können die Attributwerte (die Label für die Beziehungen, z.B. elaboration) ein- und ausgeblendet werden. Bei einer Kombination der Visualisierungstypen werden automatisch auch die Möglichkeiten der Interaktion beider Visualisierungstypen übernommen.

Die Möglichkeiten der Interaktion innerhalb eines Visualisierungstyps werden auf der visuellen Oberfläche entweder direkt am visualisierten Objekt angeordnet (z.B. die Pfeil-Schaltflächen für die Zeilenverschiebe-Aktion bei der Partitur) oder in einem gesonderten Interaktionsbereich (ebenfalls innerhalb des Visualisierungstyps) positioniert. Bei einer Kombination von Visualisierungstypen werden die Interaktionsbereiche beider Visualisierungstypen untereinander dargestellt. Der Interaktionsbereich des dominierenden Visualisierungstyps wird dabei oberhalb des Bereichs der integrierten Visualisierungsform angezeigt, letzterer wird also einfach als Block an den bestehenden Bereich „angehängt“.

Trennung der Ebenen Struktur, Stil und Interaktion

Die Ebenen Struktur, Stil und Interaktion so weit möglich voneinander getrennt werden. Die **Struktur** der Visualisierungen ist festgelegt in XHTML. Bevor überhaupt eine Transformation von XML-Daten (den im Standoff-Format gespeicherten Primärtexten und Annotationen) in das Zielformat XHTML möglich ist, muss dieses Zielformat zunächst definiert werden. Diese Definition kann nicht unabhängig von den geplanten Interaktionen erfolgen. Für deren Realisierung sind voraussichtlich zusätzliche Attribute in den XHTML-Tags notwendig und eventuell sogar zusätzliche Tags. Der **Stil** der Visualisierungen ist festgelegt in Cascading Style Sheets (CSS). Die **Interaktion** ist realisiert über DOM/Javascript (s.u.).

Modellierung der Direkten Manipulation im Entwurfsprozess

Während des Entwurfsprozesses und in anschließenden Nutzertests kann es sich heraus-

stellen, dass die Zuordnung von einer Funktionalität zu einer Nutzeraktion geändert werden muss, da sich eine andere Kombination von Aktionen als praktikabler erweist. Und nicht nur das: Auch die Nutzer selbst werden abhängig von ihren Fragestellungen und/oder ihren Aktions-Präferenzen die Zuordnung möglicherweise ändern wollen. Das bedeutet zum einen für die Gestaltung des Nutzerinterfaces, dass es einen Ort geben muss, an dem diese Zuordnung verändert werden kann, zum anderen für die technische Realisierung, dass die Zuordnung variabel ist und möglichst einfach überprüft und verändert werden kann. Für eine Formulierung von Standardeinstellungen können als Hypothese die Ergebnisse herangezogen werden, die eine formale Untersuchung der grafischen Variablen ergeben hat. Diese müssten allerdings durch geeignete Usability-Tests überprüft werden. Als technische Grundlage für eine variable Zuweisung mittels DOM/JavaScript bietet sich die in (Keith 2005) verfolgte Strategie an, eine Nutzeraktion jeweils am Anfang einer Datei zu notieren, wo sie schnell wieder aufgefunden und verändert werden kann. Die Funktionen für die verschiedenen Funktionalitäten sind voneinander unabhängig. Es wird pro Funktionalität eine Datei erstellt. Diese Datei kann bei Bedarf vom Administrator in die XHTML-Datei durch einen einfachen Verweis im `<head>` eingebunden werden.⁴⁷

Wenn diese Einbindung erfolgt ist, sind die Elemente, die in der Funktion als Auslöser der Aktion definiert sind, automatisch für die definierte Aktion „sensibilisiert“. Diese Sensibilisierung geschieht beim Laden der XHTML-Datei im Browser.

Das Problem der Segmentgröße

Eine Annotation ist charakterisiert worden über (a) ihre Bezugseinheit, (b) das Attribut, hinsichtlich dessen die Bezugseinheit beschrieben wird und (c) den zugewiesenen Wert aus der Wertemenge des Attributs. Die Bezugsebene kann der Primärtext sein, aber auch eine beliebige Annotationsebene. Diese drei Eigenschaften werden in den Basisvisualisierungen unterschiedlich abgebildet.

In der interlinearen Darstellung werden diese drei Eigenschaften abgebildet über (a) die Ausdehnung einer Tabellenzelle relativ zur Bezugsebene, (b) das Zeilenlabel und (c) den Inhalt der Zelle.

In der intralinearen Darstellung werden die drei Eigenschaften auf der visuellen Oberfläche abgebildet über (a) die Ausdehnung einer Markierung über ein oder mehrere

⁴⁷ Wenn die XHTML-Datei über XSLT erstellt wird, erfolgt das Einbinden an der entsprechenden Stelle im XSLT-Code.

Segmente der Bezugsebene hinweg; Bezugsebene ist hier primär der Primärtext, sofern dieser allerdings hinsichtlich eines anderen Attributs visuell präpariert wurde, kann mit Auswahl einer geeigneten visuellen Variable auch der Bezug zwischen den beiden Attributebenen visualisiert werden; (b) die Anzeige des ausgewählten Attributs und (c) die Anzeige des ausgewählten Wertes.

In der extralinen Darstellung werden die Eigenschaften abgebildet über (a) die Position relativ zur Bezugseinheit, (b) das Spaltenlabel und die Anzeige des Attributs und (c) die Anzeige des Wertes.

In der superlinearen Darstellung werden die drei Eigenschaften erst nach Nutzeraktion abgebildet (das scheint zunächst ein Nachteil zu sein, aber im Gegensatz zur intralinen Darstellung ist dafür eine Übersicht über alle annotierten Werte verfügbar); die Abbildung erfolgt dann wie in der intralinen Darstellung über (a) die Ausdehnung einer Markierung, (b) die Auswahl eines Attributs und (c) die Anzeige eines Wertes.

Wie die Segmentgröße für die superlineare und die intralineare Darstellung in XHTML kodiert sein kann, ist zu klären.

Zusammenfassung

Datenbanken mit reich annotierten linguistischen Daten erfordern spezifische Visualisierungen, damit diese Daten effektiv exploriert werden können. Es wurden ausgehend von einer Nutzeranalyse Anforderungen an Visualisierungen formuliert. Bekannte Tools sind auf diese Anforderungen hin bewertet worden. Möglichkeiten der Visualisierung reich annotierter linguistischer Daten sind klassifiziert und diskutiert worden. Für ausgewählte Visualisierungsformen sind Beispiele realisiert worden.

Da bei der Darstellung großer Mengen linguistischer Daten eine interaktive Komponente von zentraler Bedeutung ist, um unterschiedlichen Nutzern mit unterschiedlichen Daten und Fragstellungen spezifische Visualisierungen bieten zu können, wurde die Betrachtung dieser Komponente einer Ausformulierung in XSLT vorgezogen.

Als zentrale Ebene für die technische Realisierung hat sich die XHTML-Ebene herausgestellt, auf der zum einen die CSS-Angaben und die Interaktionsebene (Javascript) basieren, und die zum anderen das Zielformat für die XSLT-Transformation darstellt.

Es ist mittels XHTML/CSS/DOM eine interaktive webfähige Partitur realisiert worden, die es dem Nutzer erlaubt, bei feststehender Labelspalte innerhalb der Partitur zu scrollen oder an bestimmte Punkte in der Partitur zu springen, die er in einem gleichzeitig angezeigten Textfeld auswählt. In dieser Partituransicht können Zeilen verschoben, ein- und ausgeblendet und durch Hinzufügen von Leerzeilen gruppiert werden. Es können außerdem wahlweise zusätzliche Referenzzeilen eingeblendet werden, die bei der Exploration von Partituren mit sehr vielen Annotationsebenen die Orientierung unterstützen. Um einzelne Zeilen/Zeilengruppen fokussieren zu können, wurde sowohl ein temporäres als auch ein dauerhaftes Hervorheben von Zeilen realisiert.

Für eine alternative Partiturdarstellung mit Zeilenumbruch sind Möglichkeiten der Umsetzung gefunden worden. Damit ist eine Darstellung realisierbar, die sich an der Größe des Anzeigebereichs innerhalb des Nutzerinterfaces bzw. an der Größe des Ausgabebereichs beim Drucken orientiert.

Sofern Syntaxdaten vorhanden sind, können diese innerhalb der Partitur als Baum dargestellt werden. Der Nutzer kann ohne Neuladen der Seite zwischen einer platzsparenden Ebenenansicht und einer besser lesbaren Graphenansicht hin- und herwechseln. Es ist die Möglichkeit angelegt, verschiedene Teilbäume hervorzuheben sowie zusammengehörige Teilbäume zu markieren.

Ob eventuell eine verschränkte Darstellung mit SVG innerhalb der Partitur möglich ist (mit

z-index und absoluter Positionierung an zuvor per DOM bestimmten Knoten), müsste überprüft werden. Das Einbinden von SVG hätte den Vorteil, dass auch kreuzende Kanten visualisiert werden könnten (von einem ausschließlichen Gebrauch von SVG ist jedoch abzuraten, da noch nicht alle Standardbrowser SVG, XHTML, CSS und COM im Zusammenspiel unterstützen). Solange die *verschränkte* Darstellung mit SVG noch nicht gegeben ist, kann auf eine *kombinierte* Darstellung mit SVG-Grafiken zurückgegriffen werden, die nicht interaktiv sind, aber kreuzende Kanten visualisieren können.

Koreferenz in einer Partitur zu visualisieren ist bei typischerweise eher langen Texten nicht übersichtlich. Es wird daher eine intralineare Darstellung zu ermöglichen sein.⁴⁸

Um das *Problem der gleichzeitigen Darstellung heterogener Datentypen* zu lösen, ist eine integrierte Darstellung von normalen Attribut-Wert-Paaren mit RST-, Koreferenz- und Syntaxdaten konzipiert und realisiert worden.

Das Problem der *mehrfachen Annotationen* wurde gelöst durch (a) Schaffen einer besseren Orientierung durch räumliche Gruppierung von Zeilen, (b) die Möglichkeit, eine Partiturzeile temporär hervorzuheben.

Das Problem der *heterogenen Fragestellungen* wurde gelöst, indem dem Nutzer die Möglichkeit gegeben wird, die für die jeweilige Fragestellung relevanten Annotations-ebenen bzw. -ebensets ein- oder auszublenden, die Reihenfolge der Partiturzeilen selbst zu bestimmen und die visuelle Clusterung der Zeilen durch Hinzufügen oder Entfernen einer Leerzeile selbst zu erstellen (dieses Konzept mit den Leerzeilen ist noch nicht ausgereift; vermutlich ist das Konzept der Gruppe „griffiger“ als das der Leerzeile, so dass es für den Nutzer intuitiver wäre, Zeilen zu einer Gruppe hinzuzufügen, als die Gruppen durch Leerzeilen zu erstellen; dies müsste in Nutzertests überprüft werden. Zusätzlich kann je nach Relevanz der jeweiligen Daten innerhalb der Fragestellung zwischen der normalen Zeilendarstellung oder der datenspezifischen Darstellung gewechselt werden (z.B. Syntax als normale Partiturzeile vs. Syntax als Graph).

⁴⁸ Dafür gibt es an der Universität Potsdam bereits einen Prototypen.

Literaturverzeichnis

- Anderson, John Robert (2001): Kognitive Psychologie. Unter Mitarbeit von Übers. und hrsg. von Ralf Graf und Joachim Grabowski. Heidelberg: Spektrum Akad. Verl. (Reihe Spektrum-Lehrbuch).
- Berglund, Anders; Boag, Scott; Chamberlin, Don; Fernández, Mary F.; Kay, Michael; Robie, Jonathan; Siméon, Jérôme (2007): XML Path Language (XPath) 2.0. W3C Recommendation 23 January 2007. Online verfügbar unter <http://www.w3.org/TR/2007/REC-xpath20-20070123/>, zuletzt geprüft am 10.02.2007.
- Bertin, Jaques (1967): Sémiologie Graphique. Paris: Mouton/Gauthier-Villars.
- Bongers, Frank (2005): XSLT 2.0. Das umfassende Handbuch. 1. korrigierter Nachdruck. Bonn: Galileo Press.
- Bortz, Jürgen; Döring, Nicola; Bortz-Döring (2005): Forschungsmethoden und Evaluation. Für Human- und Sozialwissenschaftler ; mit 70 Tabellen. Heidelberg: Springer (Reihe Springer-Lehrbuch).
- Bos, Bert et al. (1998): Cascading Style Sheets, level 2 CSS2 Specification. W3C Recommendation (12-May-1998). REC-CSS2-19980512. Online verfügbar unter <http://www.w3.org/TR/1998/REC-CSS2-19980512/>, zuletzt geprüft am 06.02.2007.
- Bos, Bert; Lie, Hakon Wium; Lilley, Chris; Jacobs, Ian (1998): Cascading Style Sheets, level 2 CSS2 Specification. W3C Recommendation 12-May-1998. Online verfügbar unter http://www.w3.org/TR/1998/REC-CSS2-19980512, zuletzt geprüft am 10.02.2007.
- Brants, Sabine; Dipper, Stefanie; Hansen, S.; Lezius, W.; Smith, G. (2002): The TIGER Treebank: Proceedings of the Workshop on Treebanks and Linguistic Theories. Sozopol.
- Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.; Maler, Eve; Yergeau, Francois (29 September 2006): Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation 16 August 2006, edited in place 29 September 2006. Online verfügbar unter http://www.w3.org/TR/2006/REC-xml-20060816, zuletzt aktualisiert am 29 September 2006, zuletzt geprüft am 10.02.2007.
- Card, Stuart K.; Mackinlay, Jock D.; Shneiderman, Ben (Hg.) (1999): Readings in information visualization. Using vision to think. San Francisco Calif.: Morgan Kaufmann Publishers.
- Courage, Catherine; Baxter, Kathy (2005): Understanding your users. A practical guide to user requirements ; methods, tools, and techniques. Amsterdam: Morgan Kaufmann (Reihe The Morgan Kaufmann series in interactive technologies).
- DIN Deutsches Institut für Normung e.V. (Hg.) (2004): Software-Ergonomie. Stand: Januar 2004. Empfehlungen für die Programmierung und Auswahl von Software. Berlin: Beuth (Reihe DIN-Taschenbuch ; 354).
- Dipper, S. (2005): XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation: Proceedings of Berliner XML Tage 2005 (BXML). Berlin, Germany, S. 39–50.
- Dipper, S.; Götze, M.; Stede, M.: Simple Annotation Tools for Complex Annotation Tasks: an Evaluation, S. 54–62.
- Dipper, S.; Götze, M.; Stede, M.; Wegst, T. (2004): ANNIS : a linguistic database for exploring information structure: Interdisciplinary studies on information structure : ISIS ; Working papers of the SFB 632. Potsdam: Univ.-Verl, S. 245–279.
- Dybkjaer, Laila; Berman, Stephen; Bernsen, Niels Ole (July 2001): Requirements Specification for a Tool in Support of Annotation of Natural Language Interaction and Multimodal Data.
- Eberleh, Edmund; Oberquelle, Horst; Oppermann, Reinhard (Hg.) (1994): Einführung in die Software-Ergonomie. 2., völlig neu bearb. Aufl. Gestaltung graphisch-interaktiver Systeme ; Prinzipien, Werkzeuge, Lösungen. Berlin u.a.: Gruyter.
- Eibl, Maximilian (2003): Visualisierung im Document Retrieval. Theoretische und praktische Zusammenführung von Softwareergonomie und Graphik Design. Bonn: Informationszentrum Sozialwiss. (Reihe Forschungsberichte / Informationszentrum Sozialwissenschaften, 7).
- Evert, Stefan; Carletta, Jean; O'Donnell, Timothy J.; Kilgour, Jonathan; Vögele, Andreas; Voormann, Holger: The NITE Object Model. Version 2.1 (24 March 2001). NITE Internal Document. Online verfügbar unter <http://www.ltg.ed.ac.uk/NITE/documents.html>, zuletzt geprüft am 20.08.2006.
- Ferraiolo, Jon; Fujisawa, Jun; Jackson, Dean (2004): Scalable Vector Graphics (SVG) 1.1 Specification. W3C Recommendation 14 January 2003. Online verfügbar unter <http://www.w3.org/TR/2003/REC-SVG11-20030114/>, zuletzt geprüft am 10.02.2007.

- Glaser, Wilhelm R. (1994): Menschliche Informationsverarbeitung. In: Eberleh, Edmund; Oberquelle, Horst; Oppermann, Reinhard (Hg.): Einführung in die Software-Ergonomie. Gestaltung graphisch-interaktiver Systeme ; Prinzipien, Werkzeuge, Lösungen. 2., völlig neu bearb. Aufl. Berlin u.a.: <de> Gruyter. Mensch - Computer - Kommunikation : Grundwissen ; 1, S. 7–52.
- Glück, Helmut (1993): Metzler Lexikon Sprache. Stuttgart ;Weimar: Metzler.
- Götze, M. (2003): Zur Annotation von Informationsstruktur. Diplomarbeit. Potsdam. Universität Potsdam, Institut für Linguistik.
- Grewendorf, Günther; Hamm, Fritz; Sternefeld, Wolfgang (1998): Sprachliches Wissen. Eine Einführung in moderne Theorien der grammatischen Beschreibung. 10. Auflage. Frankfurt am Main: Suhrkamp (Reihe Suhrkamp Taschenbuch Wissenschaft).
- Grosso, Paul; Maler, Eve; Marsh, Jonathan; Walsh, Norman (2003): XPointer Framework. W3C Recommendation 25 March 2003. Online verfügbar unter <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>, zuletzt geprüft am 10.02.2007.
- Jannedy, S.; Mendoza-Denton, N. (2005): Structuring Information through Intonation and Gesture. In: Ishihara, S.; Schmitz, M.; Schwarz, A. (Hg.): Approaches and Findings in Oral, Written and Gestural Language, Interdisciplinary Studies on Information Structure (ISIS). Potsdam: Universitätsverlag Potsdam., Bd. 3, S. 199–244.
- Johnson, Jeff (2000): GUI bloopers. Don'ts and do's for software developers and web designers. San Francisco: Morgan Kaufmann Publ. (Reihe The Morgan Kaufmann series in interactive technologies).
- Joshi, A.; Webber, B. (2004): Penn Discourse Treebank - Annotation Tutorial. Online verfügbar unter <http://www.seas.upenn.edu/~pdtb/manual/pdtb-tutorial.pdf>, zuletzt geprüft am 21.02.2007.
- Kay, Michael (2007): XSL Transformations (XSLT) Version 2.0. W3C Recommendation 23 January 2007. Online verfügbar unter <http://www.w3.org/TR/2007/REC-xslt20-20070123/>, zuletzt geprüft am 10.02.2007.
- Keith, Jeremy (2005): DOM Scripting. Web Design with JavaScript and the Document Object Model. New York: Springer.
- Keller, Peter R.; Keller, Mary M. (1993): Visual cues. Practical data visualization. Los Alamitos Calif.: IEEE Computer Soc. Press.
- Kennedy, Graeme D. (1998.): An introduction to corpus linguistics. London ;New York: Longman (Reihe Studies in language and linguistics).
- Le Hors, Arnaud; Le Hégarret, Philippe; Wood, Lauren; Nicol, Gavin; Robie, Jonathan; Champion, Mike; Byrne, Steve (2000): Document Object Model (DOM) Level 2 Core Specification Version 1.0. W3C Recommendation 13 November, 2000. Online verfügbar unter <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113>, zuletzt geprüft am 10.02.2007.
- MacEnery, Tony; Wilson, Andrew (2001): Corpus linguistics. An introduction. Edinburgh: University Press (Reihe Edinburgh textbooks in empirical linguistics).
- Malhotra, Ashok et al.: XQuery 1.0 and XPath 2.0 Functions and Operators. W3C Proposed Recommendation 21 November 2006. Online verfügbar unter <http://www.w3.org/TR/xpath-functions/>, zuletzt geprüft am 21.01.2007.
- Mann, William C.; Thompson, Sandra A. (June 1987): Rhetorical Structure Theory: A Theory of Text Organization. University of Southern California, Information Sciences Institute (ISI). Report number ISI/RS-87-190; NTIS Identifying Number ADA 183038)
- Müller, Christoph (01.02.2005): MMAX2 Annotation Tool - Quick Start Guide. EML Research gGmbH
- Müller, Christoph (01.02.2005): MMAX2 Annotation Tool - Style Sheet Guide. EML Research gGmbH
- Müller, Christoph (2006): MMAX2. Version 2 (beta1): EML Research gGmbH. Online verfügbar unter <http://mmax.eml-research.de>, zuletzt geprüft am 27.11.2006.
- Müller, Christoph; Strube, Michael (2001): Annotating Anaphoric and Bridging Relations with MMAX: Proceedings of the Second SIGdial Workshop on Discourse and Dialogue. Aalborg, Denmark: Association for Computational Linguistics, Bd. 16, S. 1–6.
- North, Chris; Shneiderman, Ben (15-Oct-1998): A Taxonomy of Multiple Window Coordinations. Herausgegeben von UM Computer Science Department. CS-TR-3854; UMIACS-TR-97-83)
- O'Donnell, Michael: RSTTool -- an RST Markup Tool. Online verfügbar unter <http://www.wagsoft.com/RSTTool/>,

zuletzt geprüft am 25.01.2007.

O'Donnell, Michael: RSTTool 2.4 -- A Markup Tool for Rhetorical Structure Theory: Proceedings of the International Natural Language Generation Conference (INLG'2000). 13-16 June 2000, Mitzpe Ramon, Israel, S. 253–256.

O'Donnell, Mick (Juni 2004): RSTTool. Version 3.45: WagSoft systems. Online verfügbar unter <http://www.wagsoft.com/RSTTool/index.html>, zuletzt geprüft am 25.01.2007.

Pittner, Karin; Berman, Judith (2004): Deutsche Syntax. Ein Arbeitsbuch. Tübingen: Narr (Reihe Narr Studienbücher).

Plaisant, Catherine (2004): The challenge of information visualization evaluation: AVI 2004. Proceedings of the working conference on Advanced visual interfaces, May 25 - 28, 2004, Gallipoli, Italy: ACM Press, S. 109–116.

Preim, Bernhard (1999): Entwicklung interaktiver Systeme. Grundlagen, Fallbeispiele und innovative Anwendungsfelder ; mit 20 Tabellen. Berlin: Springer.

Reitter, David; Stede, Manfred (2003): Step by step: Underspecified markup in incremental rhetorical analysis. Budapest , Proc. of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)).

Sarodnick, Florian; Brau, Henning (2006): Methoden der Usability Evaluation. Wissenschaftliche Grundlagen, praktische Anwendung. Bern: Huber (Reihe Psychologie Praxis).

Schmidt, Thomas: EXMARaLDA Partitur-Editor. Version 1.3.3. Online verfügbar unter <http://www1.uni-hamburg.de/exmaralda/>, zuletzt geprüft am 12.02.2007.

Schmidt, Thomas (2003): Visualising Linguistic Annotation as Interlinear Text. Universität Hamburg. (Arbeiten zur Mehrsprachigkeit, Serie B, 46)

Schmidt, Thomas (2005): Computergestützte Transkription. Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln. Frankfurt am Main: Lang (Reihe Sprache, Sprechen und Computer, 7).

Schneiderman, Ben; North, Chris: A Taxonomy of Multiple Window Coordinations. UM Computer Science Department. CS-TR-3854)

Schumann, Heidrun; Müller, Wolfgang (2000): Visualisierung. Grundlagen und allgemeine Methoden. Berlin: Springer.

Shneiderman, Ben; Plaisant, Catherine (2005): Designing the user interface. Strategies for effective human-computer interaction. Boston: Addison-Wesley.

Stede, Manfred (2004): The Potsdam Commentary Corpus: Proceedings of the ACL-04 Workshop on Discourse Annotation. Barcelona, Spain.

The Stuttgart-Tübingen Tagset (STTS). Online verfügbar unter <http://www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts.html>, zuletzt geprüft am 20.02.2007.

Toolbox. Field Linguist's Toolbox. Version 1.5.0 Aug 2006: SIL International. Online verfügbar unter <http://www.sil.org/computing/toolbox/index.htm>, zuletzt geprüft am 28.01.2007.

Tufte, Edward R. (2005): Envisioning information. Cheshire Conn.: Graphics Press.

Ware, Colin (2004): Information visualization. Perception for design. 2., aktualisierte und erweiterte Auflage. San Francisco: Morgan Kaufman.

Anhang

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum Unterschrift

Lebenslauf

Sommer 1996	1. Staatsexamen Humanmedizin an der Otto-von-Guericke-Universität Magdeburg
8/1996 – 6/1997	Studium der Humanmedizin an der Université Louis Pasteur Strasbourg (Frankreich)
1997/98	Studium der Humanmedizin an der Humboldt-Universität zu Berlin (Charité)
seit 1998	(mit Unterbrechungen) Studium der Germanistischen Linguistik und Romanistik an der Humboldt-Universität Berlin, Studium der Kommunikationswissenschaft an der Technischen Universität Berlin
11/2000 - 12/2001	Studentische Mitarbeiterin am Institut für Wirtschafts- und Erwachsenenpädagogik (Abteilung Pädagogik und Informatik) der Humboldt-Universität zu Berlin
10/2002 - 2/2005	Studentische Mitarbeiterin am Multimedia Lehr- und Lernzentrum der Humboldt-Universität zu Berlin
seit 1/2006	Studentische Mitarbeiterin im Cornelsen Verlag Berlin (Design Digitale Medien)