Hypertextsorten-Klassifikation als Grundlage generischer Informationsextraktion

Georg Rehm

1 Einleitung

Dieser Beitrag stellt technologische Aspekte des Projekts Hypnotic (*Hypertexts and their Organisation into a Taxonomy by means of Intelligent Classification*) vor, in dem der Ansatz verfolgt wird, mit texttechnologischen und computerlinguistischen Verfahren HTML-Dokumente abstrakten *Hypertextsorten* zuzuordnen (vgl. einleitend hierzu Rehm, in diesem Band sowie Rehm, 2002b). In einem zweiten Schritt sollen – basierend auf dem Wissen, dass eine bestimmte Hypertextsorte vorliegt – generische Informationsextraktionsprozesse ausgeführt werden, um gezielt auf atomare und modulare Informationseinheiten, die in einem Dokument enthalten sind, zugreifen zu können. Insgesamt wird also eine Hypertextsorten-getriebene Konvertierung arbiträrer HTML-Dokumente der Untersuchungsdomäne in korrespondierende XML-Formate angestrebt.

Die eingangs angesprochenen technologischen Aspekte betreffen zwei wesentliche Bausteine des Hypnotic-Projekts: In Abschnitt 2 werden zunächst die Motivation für den Aufbau einer Korpusdatenbank, die aus ca. 4 Mio. deutschsprachigen Dokumenten deutscher Hochschulen besteht, sowie ihre technische Umsetzung erläutert. Abschnitt 3 geht daraufhin auf die Architektur des Analyse-Systems sowie den Ansatz zur generischen Informationsextraktion ein.

2 Die Hypnotic-Korpusdatenbank

Zur Identifikation verschiedener Hypertextsorten wurden im Kontext der *digital genre*- bzw. *web genre*-Diskussion verschiedene Studien durchgeführt, in denen versucht wurde, Dokumenten, die mittels einer zufälligen Dokumentauswahl von großen Suchmaschinen wie z.B. Altavista bezogen wurden, Hypertextsorten zuzuordnen (beispielsweise in Dillon/Gushrowski 2000, Roussinov *et al.* 2001, Shepherd/Watters 1999, Crowston/Williams 2000, Haas/Grams 2000). Da in diesen Studien keine weitere Eingrenzung bzgl. der Dokumentsammlung vorgenommen wurde, sind die Resultate sowohl sehr unterschiedlicher als auch sehr genereller Natur, so dass sie für Zwecke der maschinellen Klassifikation von Hypertextsorten nicht verwendet werden können. Die Arbeiten legen nahe, dass eine systematische Untersuchung von Hypertextsorten zunächst nur mit Hilfe von Einschränkungen bzgl. des Untersuchungsgegenstandes vorgenommen werden kann. Hierzu wurden in Hypnotic die deutschsprachigen Webseiten

der deutschen Hochschulen ausgewählt, da in dieser thematischen Domäne ein sehr hohes Maß an Strukturiertheit vorzufinden ist (vgl. etwa Müller 1999).
Da sowohl das Layout als auch der Inhalt von HTML-Dokumenten sehr häufig aktualisiert werden (Koehler 2002), war es notwendig, einen Schnappschuss (Walker 1999, Hawking *et al.* 1999) der Untersuchungsdomäne anzufertigen, es wurde also eine Art lokales Archiv erzeugt, das die deutschsprachigen Dokumente der deutschen Hochschulen umfasst. Zu diesem Zweck wurden die Webserver sämtlicher Hochschulen mit Hilfe eines Webcrawlers rekursiv traversiert, um sukzessive alle verfügbaren deutschsprachigen HTML-Dokumente zu sammeln und auf dem lokalen Server zu spiegeln. Tabelle 1 zeigt den Inhalt sowie den Umfang der Hypnotic-Korpusdatenbank und macht verschiedene Angaben zur Anzahl aller Webseiten aller deutschen Universitäten.

Während der Datensammlung werden durch Tests, die sowohl intern im Webcrawler, als auch extern in nachgeschalteten Werkzeugen durchgeführt werden, umfangreiche Beschränkungen realisiert. Zunächst werden lediglich diejenigen Dateien, die bestimmten Datei- bzw. Medientypen (vgl. Tab. 1 sowie generell RFC 2616, Fielding *et al.* 1999) entsprechen, im Korpus abgelegt. Da binäre Dateitypen für das Projekt keine entscheidende Relevanz besitzen und unverhältnismäßig viel Festplattenplatz in Anspruch nähmen (vgl. die Angabe zur „Gesamtgröße aller entfernten Webserver“ in Tab. 1), werden entsprechende Verweise (etwa innerhalb eines `<img>` Elements), auf die entfernten Server umgeschrieben. Zentral ist die Überprüfung der Sprache, in der ein HTML-Dokument verfasst worden ist. Der in Perl implementierte, Lexikon-basierte Sprachenidentifizierer (allgemein hierzu etwa Langer 2001) `germanp.pl` führt zunächst eine Tilgung des HTML-Markups durch, um nach einer rudimentären Tokenisierung mit Hilfe einer Heuristik, die neben der reinen Trefferanzahl auch die Wortlänge berücksichtigt, das Verhältnis der deutschsprachigen und der unbekannten Token zu ermitteln. Liegt diese Angabe oberhalb eines bestimmten Schwellwerts, wird als Sprache „Deutsch“ angenommen und das Dokument im Korpus abgelegt (vgl. Cowie *et al.* 1998 für einen ähnlichen Ansatz). Dieses Verfahren arbeitet mit einer Präzision von 96,6%. Weitere Informationen zum Sprachenidentifizierer und zu technischen Aspekten der Korpusdatenbank befinden sich in Rehm (2001).

Der Kern der Korpusdatenbank besteht aus drei Bausteinen: In einer relationalen Datenbank (eingesetzt wird MySQL unter Linux, vgl. DuBois 1999) werden fast alle Metadaten abgelegt, die von den besuchten Webservern für jede angefragte Datei in Form von HTTP Response Headern (Fielding *et al.* 1999) zurückgeliefert werden, wodurch umfangreiche Retrieval-Funktionen ermöglicht werden. Die Dokumente selbst werden aus Performanzgründen nicht in der Datenbank, sondern im UNIX-Dateisystem gespeichert; in einer Datenbanktabelle befindet sich lediglich eine voll spezifizierte Pfadangabe für jede im Korpus enthaltene Datei. Der Datenbankserver ist mit einem Webserver aus-

gestattet, und die Korpusdateien befinden sich in einem Bereich, auf den der Webserver Zugriff hat. Auf diese Weise ist es mit Hilfe eines kapselnden Perl-Moduls möglich, über das Netzwerk eine SQL-Query an die Datenbank zu richten, eine Menge von Dokument-Kennungen zu erhalten und diese wiederum zu benutzen, um die HTML-Dokumente für eine Weiterverarbeitung mit Hilfe von HTTP auf ein autarkes Analysesystem zu transferieren.

Tabelle 1: Inhalt und Umfang der Hypnotic-Korpusdatenbank

Universitäten in der Hypnotic-Korpusdatenbank:	100
* Allgemeine Universitäten (*vollständig*)	62
* Technische Hochschulen (*vollständig*)	12
* Musik- und Kunsthochschulen (*partiell*)	5
* Wirtschaftshochschulen (*partiell*)	5
* Sonstige Hochschulen (*partiell*)	16
Dauer der Datensammlung:	16.01.2001-07.09.2002
Traversierte Webserver insgesamt:	14 968
Auf Port 80 operierende Webserver:	13 885
Anzahl per HTTP erreichbarer Dateien:	16 196 511
Anzahl HTML-Dokumente:	8 465 105
Gesamtgröße aller entfernten Webserver:	701 464,29 MB
Gesamtgröße der Korpusdatenbank:	40 914,99 MB
Laufende Wortformen (gesamt; nur bezogen auf Dokumente vom Typ `text/html`):	1 138 794 715
Laufende Wortformen (eindeutig; nur bezogen auf Dokumente vom Typ `text/html`):	12 120 162
Dateien im Korpus gesamt:	4 294 417
* Dateien vom Medientyp `text/html`:	3 956 692
* Dateien vom Medientyp `text/plain`:	270 400
* Dateien vom Medientyp `text/css`:	35 651
* Dateien vom Medientyp `text/xml`:	25 871
* Dateien vom Medientyp `text/sgml`:	956
* Dateien vom Medientyp `message/news`:	490
* Dateien vom Medientyp `message/rfc822`:	436

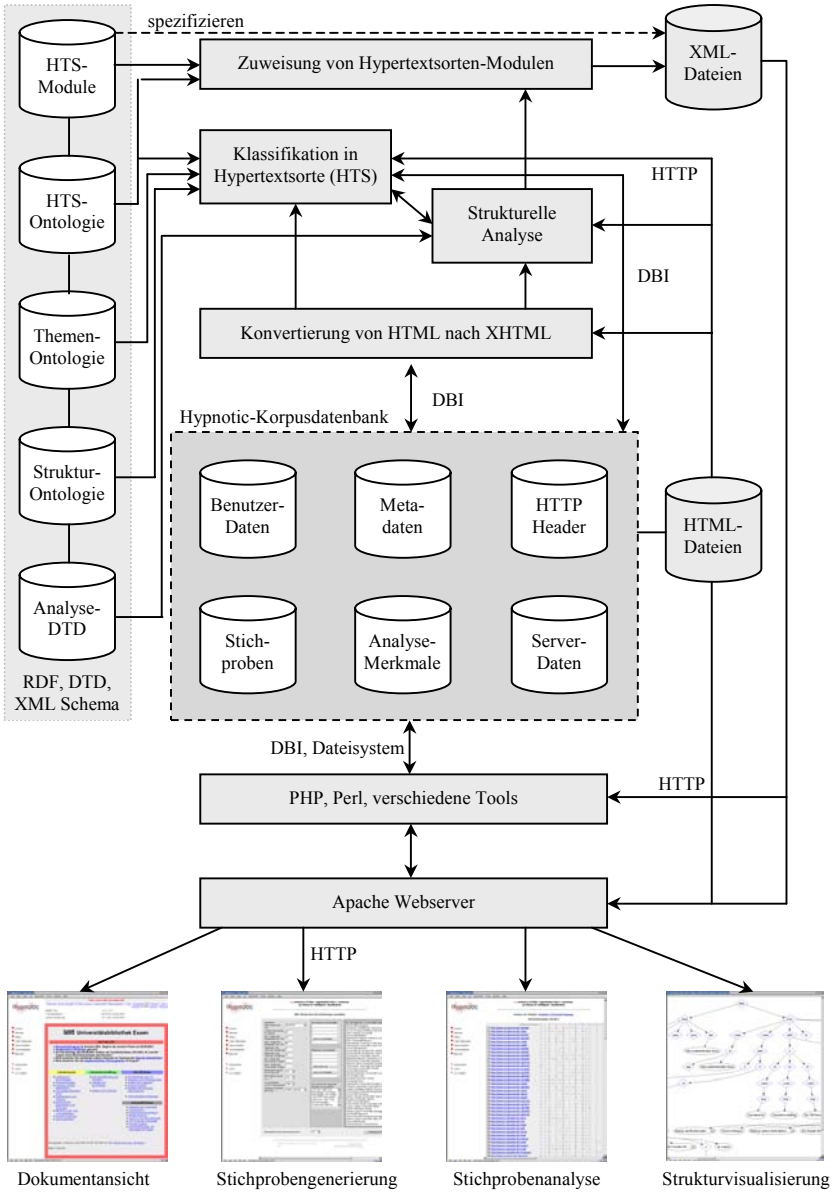

Abbildung 1: Arc2, das primär aus den Bestandteilen Zugangs- und Analyseoberfläche, Korpusdatenbank sowie maschinellen Analysemodulen besteht

Neben dieser indirekten Zugriffsmethode, die für das eigentliche Hypnotic-Analysesystem entwickelt wurde, existiert als dritter Baustein auch eine Web-Schnittstelle, die für den manuellen Zugriff auf das Korpus konzipiert wurde. Über dieses in der Skripting-Sprache PHP realisierte Web-Frontend können zahlreiche Funktionen durchgeführt werden. Hierzu gehört die Suche, Betrachtung und Visualisierung von Dokumenten sowie ihrer Baumstruktur, die Generierung, Analyse und Darstellung von Stichproben, die Exploration des Korpus, eine Benutzerverwaltung sowie statistische Informationen des Korpusdatenbank-Servers. Mit Hilfe des Systems wurden bislang drei Studien durchgeführt, die sprachliche Eigenschaften von persönlichen Homepages untersuchen (Bayerl 2002, Rehm 2002a, Rehm 2003a). Abbildung 1 zeigt die geschilderten direkten und indirekten Zugriffsmöglichkeiten auf die Korpusdatenbank sowie die vereinfachte Tabellenstruktur.

3 Hypertextsorten-basierte Informationsextraktion

Das klassische Anwendungsszenario der Informationsextraktion ist die Ermittlung spezifischer Informationsteile, z.B. aus Nachrichtenmitteilungen zu terroristischen Anschlägen, um hiermit spezielle Templates zu füllen, die generelle Informationen des jeweiligen Texttyps spezifizieren (vgl. Cowie/Wilks 2000). Im World Wide Web hingegen ist eine Ausprägung der Informationsextraktion vorherrschend, die als *Wrapping* bezeichnet wird (vgl. Rehm 2003b für einen Überblick). Wrapper werden mit Regeln ausgestattet, die es ihnen ermöglichen, Informationen aus Webseiten zu extrahieren, um sie daraufhin als XML-Datei oder Datenbankeintrag zu exportieren. Hierzu findet insbesondere eine Analyse der Strukturierung eines Dokuments auf der Ebene der HTML-Auszeichnung statt, d.h. ein Wrapper ermittelt Informationsbestandteile aufgrund ihrer Position in dem Baum, der von den HTML-Elementen aufgespannt wird. Wrapper werden entweder manuell[1] oder halbautomatisch[2] mit Extraktionsregeln versorgt, oder sie werden – mit Hilfe induktiver Lernverfahren – von dem System vollautomatisch akquiriert (vgl. die Übersicht in Eikvil 1999). HTML-Dokumente, die im Sinne von Hsu/Dung (1998) strukturiert (etwa dynamisch aus Datenbankinhalten generierte Webseiten) oder semistrukturiert (s.o., jedoch mit strukturellen Abweichungen) aufgebaut sind, können auf diese Weise relativ robust verarbeitet werden. Probleme bereiten den verschiedenen Wrapping-Ansätzen vor allem diejenigen Dokumente, die manuell erzeugt werden, da deren interne Struktur nicht notwendigerweise über Regularitäten verfügt. Ein

1 Myllymaki (2001) setzt hierzu beispielsweise manuell erstellte XSLT-Stylesheets ein, die auf XHTML-Dateien angewendet werden, um Informationen zu extrahieren.

2 Hierbei visualisieren graphische Werkzeuge die Baumstruktur eines zu verarbeitenden Webseitentyps und ermöglichen somit die Generierung funktionsfähiger Wrapper ohne eine explizite Implementierung von Regelsystemen (Sahuguet/Azavant 2001, Liu *et al.* 2000).

zentrales Problem ist die mangelnde Adaptivität von Wrappern, da Layoutänderungen der Quelldokumente zwangsläufig auch Änderungen der zugrundeliegenden HTML-Struktur bewirken, wodurch Wrapper nicht mehr in der Lage sind, die gewünschten Informationen zu lokalisieren.

Im Hypnotic-Analysesystem wird zur Realisierung von Informationsextraktionsprozessen ein abstrakterer Ansatz verfolgt, der primär auf einer generischen, strukturellen Analyse von HTML-Dokumenten sowie den Ergebnissen der Komponente zur Hypertextsorten-Klassifikation basiert (vgl. hierzu Rehm, in diesem Band). Im Folgenden werden die bereits implementierten und noch in der Planung befindlichen Vorverarbeitungs-, Analyse- und Annotationsmodule skizziert, woraufhin ein Beispiel die generelle Vorgehensweise schildert. Dabei ist das Hauptziel, für gegebene HTML-Dokumente einer bestimmten Hypertextsorte möglichst explizite XML-Annotierungen der vorgefundenen Inhalte zu generieren.

3.1 Konvertierung arbiträrer HTML-Strukturen nach XHTML

Die Verarbeitung der in der Hypnotic-Korpusdatenbank enthaltenen Dateien beginnt mit einem Konvertierungsschritt, der beliebige HTML-Dateien nach XHTML Transitional (Pemberton 2002) überführt (vgl. Abb. 1), um den sich anschließenden Analyse- und Transformationskomponenten wohlgeformte – jedoch nicht notwendigerweise valide – XML-Instanzen zur Verfügung stellen zu können. Mit der Wahl von XHTML als kanonischem Zwischenformat ist somit gewährleistet, dass Tools, Standards und Visualisierungsverfahren eingesetzt werden können, die für den Umgang mit XML-Dateien entwickelt wurden (z.B. unterschiedliche XML-Parser oder Implementationen von DOM oder XPath).

Ein Kernziel des Hypnotic-Systems soll es sein, *beliebige* HTML-Strukturen robust verarbeiten zu können. Aus diesem Grund wurde die für die weiteren Verfahren entscheidende initiale Konvertierung nach XHTML als zweistufiger Prozess realisiert: Ein Perl-Modul, das sowohl von der Web-Oberfläche als auch von den Analysekomponenten eingesetzt wird, verkapselt dabei eine Verarbeitungsheuristik, die zum einen eine modifizierte Version des Werkzeugs tidy (http://tidy.sourceforge.net) und zum anderen das im Comprehensive Perl Archive Network (CPAN, http://www.cpan.org) erhältliche Modul HTML::TreeBuilder ansteuert.[3] Eine wesentliche Funktion von tidy stellt die Möglichkeit dar, HTML-Dokumente nach XHTML zu konvertieren. Sowohl tidy als auch HTML::TreeBuilder implementieren weiterhin Regeln für einen robusten

3 Im Zuge der Auswahl der Werkzeuge wurden auch Alternativen getestet, etwa die Perl-Module XML::PYX sowie XML::Driver::HTML und das UNIX-Tool recode, die jedoch keine zufriedenstellenden Resultate liefern.

Umgang mit fehlerhaften HTML-Strukturen, die im World Wide Web sehr häufig anzutreffen sind, da praktisch alle Browser über fehlertolerante Parsing-Strategien für – aus Sicht der verschiedenen mit Hilfe von SGML spezifizierten HTML-Dokumenttypdefinitionen – ungültigen HTML-Code verfügen. Da tidy diesen Prozess mit einer deutlich höheren Präzision ausführt, wird zunächst versucht, das HTML-Dokument mit Hilfe von tidy einzulesen und nach XHTML zu konvertieren. Gelingt dies nicht, wird HTML::TreeBuilder als Fallback-Komponente eingesetzt, um ein fehlerfreies HTML-Dokument zu erzeugen, das daraufhin mittels tidy nach XHTML konvertiert wird. Eine Evaluierung mit 10000 zufällig ausgewählten HTML-Dokumenten zeigt, dass auf diese Weise 98,7% aller Dateien korrekt konvertiert werden, wobei in 270 Fällen die Hilfestellung von HTML::TreeBuilder benötigt wird. Die Wohlgeformtheit der entstandenen X(HT)ML-Dokumente wurde daraufhin mit dem nicht validierenden XML-Parser expat (in Form des Perl-Moduls XML::Parser) überprüft. Lediglich fünf der 9872 erfolgreich konvertierten Dokumente erzeugen dabei eine Fehlermeldung, die in fast allen Fällen auf enthaltene Binärzeichen zurückzuführen sind, die einen Konflikt mit dem deklarierten Unicode-Zeichensatz UTF-8 hervorrufen. Abschließend wurde versucht, die Dokumente mit dem XML/SGML-Parser onsgmls[4] gegen die XHTML 1.0 Transitional DTD zu validieren. Fehler, die sich auf die Schachtelung einzelner Elemente beziehen, treten kaum auf; das Gros der Fehlermeldungen resultiert aus falschen Attributwerten (sehr häufig beispielsweise bei `valign`) oder nicht existenten Elementen (z.B. `<blink>` oder `<spacer>`).

3.2 Strukturanalyse von XHTML-Dokumenten

Sobald ein aus der Korpusdatenbank stammendes HTML-Dokument nach XHTML konvertiert ist, kann es mit XML-Technologien weiterverarbeitet werden. Das Modul zur strukturellen Analyse (vgl. Abb. 1), das derzeit in Perl implementiert wird, analysiert zahlreiche Eigenschaften der XHTML-Dokumente. Für diesen Zweck wird primär das Perl-Modul XML::LibXML[5] eingesetzt, das u.a. eine vollständige Implementierung des Document Object Model Level 2 (DOM, Hors *et al.* 2000) enthält; DOM erlaubt eine baumbasierte Verarbeitung von XML-Daten, die innerhalb dieser speziellen Implementierung von einem XPath-Prozessor (Clark/DeRose 1999) begleitet wird, um einzelne Knoten und Teilbäume lokalisieren zu können.

4 Hierbei handelt es sich um eine stark erweiterte Version des Parsers nsgmls, der wiederum Bestandteil des SP-Pakets von James Clark ist, siehe http://openjade.sourceforge.net und http://www.jclark.com/sp/.

5 Derzeit verfügbar in der Version 1.54; dieses benutzt die C Bibliothek libxml (Version 2.4.26), die im Rahmen der Entwicklung der Desktop-Oberfläche Gnome entstanden ist, siehe http://xmlsoft.org.

Die Analyse der Strukturen beginnt mit dem Wurzelelement des XHTML-Dokuments. Ineinander verschaltete, rekursive Funktionen berechnen im Folgenden verschiedene Eigenschaften für jedes Element des Strukturbaums. Hierzu gehören derzeit vor allem diejenigen Merkmale, die sich auf das konkrete Markup beziehen, etwa die vollständige Anzahl der Kind-Elemente, der prozentuale Anteil der Elemente eines Teilbaums im Verhältnis zum Gesamtbaum sowie vergleichbare Angaben für die Anzahl der Wörter, die in den Textknoten eines Teilbaums enthalten sind. Hyperlinks werden bzgl. ihrer Linkziele analysiert, die derzeit als `external` (Webserver in anderer Domain), `samedomain` (anderer Server in gleicher *second level*-Domain, falls etwa ein Link von www.uni-giessen.de nach opac.uni-giessen.de vorliegt) sowie internal (gleicher Webserver) kategorisiert werden. Graphik- und Bilddateien, die mittels des Elements `<img>` referenziert werden, werden zunächst auf ihre Verfügbarkeit untersucht und bei einer positiven Rückmeldung des entfernten Webservers in das Analysesystem übertragen. Von diesen Dateien werden derzeit lediglich die physikalischen Abmessungen ermittelt, wobei explizite Angaben mit Hilfe der Attribute `height` und `width` des Elements `<img>` eine höhere Präzedenz besitzen.[6]

Da die Verarbeitung auf einer mehrfach rekursiven Traversierung des Elementbaums bzw. der DOM-Repräsentation eines Dokuments basiert, werden Analyseergebnisse unmittelbar innerhalb dieser Datenstruktur abgelegt. Zu diesem Zweck wird bei Beginn der Analyse in dem Knoten-Objekt, das das Wurzelelement `<html>` repräsentiert, ein eigener Namensraum (Bray *et al.* 1999) mit dem Präfix `hypnotic:` deklariert (zusätzlich zum Default-Namespace von XHTML 1.0), der in Abbildung 1 als „Analyse-DTD“ bezeichnet wird. Analyseergebnisse können nun jeweils als Attribute dieses speziellen Namespaces unmittelbar in bestehende (X)HTML-Elemente eingetragen werden. Umfassendere Strukturen werden durch spezielle XML-Elemente des Hypnotic-Namespaces in der DOM-Repräsentation markiert. Mit Hilfe dieser synchronen Auszeichnung wird der Einsatz von XML-Techniken sowohl für die zu analysierenden Daten als auch für Analyseergebnisse ermöglicht.[7] Neben dem unmittelbaren Zugriff auf bereits ermittelte Analyseergebnisse noch während der Verarbeitung besteht ein weiterer Vorteil darin, zu jedem Zeitpunkt

6 Hierzu wird das Perl-Modul Image::Size benutzt, das wiederum verschiedene UNIX-Werkzeuge für diese Aufgabe kapselt. Geplant ist, auch den tatsächlichen Inhalt von Grafikdateien in die Analyse einfließen zu lassen, beispielsweise die Anzahl der Farben oder ob Transparenz vorliegt, um etwa eine Aussage darüber treffen zu können, ob es sich bei einer gegebenen Datei eher um eine Strichzeichnung oder ein Photo handelt (hierzu auch Asirvatham/Ravi 2001). Erkenntnisse aus dem Bereich des Dokumentverstehens und der Dokumentanalyse könnten für diese Aufgabe ebenfalls relevant sein, vgl. u.a. Shin *et al.* (2001).

7 Durch diese Anreicherung eines Webdokuments um Analyseinformationen vergrößert sich ein (X)HTML-Dokument zurzeit etwa um den Faktor 20.

wohlgeformte XML-Dateien ausgeben zu können, die daraufhin gefiltert oder visualisiert werden können.

Das Ziel der Analyse ist, den meist sehr komplexen (X)HTML-Elementbaum in abstraktere, logische Strukturen zu partitionieren[8], damit diese von den nachfolgenden Komponenten (insbesondere der Zuordnung zu Hypertextsorten-Modulen) verarbeitet werden können. Durch die Analyse wird eine Art Meta-Ansicht auf den Elementbaum gelegt, die durch die Elemente und Attribute des Analyse-Namensraumes repräsentiert wird. Die Informationen, die hierdurch nach Fertigstellung der Algorithmen zur Strukturanalyse vorliegen sollen, umfassen vor allem grobe textuelle und makrostrukturelle Einheiten, z.B. Überschriften, Listen, Textabschnitte, Trennelemente oder interaktive Bereiche. Der eigentliche Zweck von (X)HTML ist prinzipiell, eben solche Strukturen auszeichnen zu können, jedoch werden derartige Informationseinheiten de facto auf vielfältige Weise realisiert. Eine Liste sollte zwar mit Hilfe der Elemente `<ul>` (*unnumbered list*), `<ol>` (*ordered list*) oder `<dl>` (*definition list*) ausgezeichnet werden, jedoch finden sich auch häufig lediglich einzelne Textabschnitte (`<p>` oder `<br>`), die nur wenige Worte umfassen und auf der linken Seite von einer kleinformatigen Graphik, die einen *bullet point* darstellt, flankiert sind. Aus diesem Grund muss dieses Analysemodul zwangsläufig über sehr umfangreiche und robuste Möglichkeiten verfügen, die grundlegenden textstrukturellen Merkmale eines HTML-Dokuments erkennen und entsprechend annotieren zu können. Das Element `<hr>` (*horizonal rule*) beispielsweise erzeugt einen horizontalen Strich und wird von Autoren praktisch immer als räumlicher Trenner einzelner Informationsteile benutzt. Hierzu kann jedoch auch eine Graphik benutzt werden, deren Vorkommen sich mittels einer Analyse ihrer Abmessungen ermitteln lässt, woraufhin ein solches Vorkommen – ebenso wie das Element `<hr>` – mit dem Attribut-Wert-Paar `hypnotic:TagGroup="separator"` markiert wird.[9] Auf diese Weise können auch Werbebanner (die standardisierte Abmessungen wie etwa 468x60 Punkte besitzen) und Icons (häufig 32x32 oder 16x16) entsprechend ausgezeichnet werden. Zwei weitere wichtige Werte für das Attribut `TagGroup` sind `inline` und `block`. Diese beiden Gruppen werden bereits in den HTML-Dokumenttypdefinitionen unterschieden und umfassen einerseits Elemente, die sich nur lokal auf einzelne Wörter eines größeren Blocks beziehen (`<strong>`, `<em>`, `<u>` etc.) und andererseits diejenigen Elemente, die ihrerseits größere Blöcke konstituieren (`<p>`, `<code>`, `<table>` etc.).

8 Vergleichbare Ansätze werden in DiPasquo (1998), Chan/Yu (1999), Carchiolo *et al.* (2000) und Chen *et al.* (2001) dargestellt.

9 Hier wird der Quotient aus Breite und Höhe berechnet; Werte ≥ 10 werden als `separator` angesehen.

Zur Vervollständigung des Moduls zur Strukturanalyse werden zahlreiche wietere Funktionen implementiert, beispielsweise eine Satzgrenzenerkennung, die Bestimmung der logischen Grenzen einer Gruppe von HTML-Dokumenten etc. Auch eine Betrachtung der eingesetzten Schriftgrößen ist notwendig, da Überschriften häufig nicht mit den hierfür vorgesehenen Elementen `<h1>` bis `<h6>`, sondern explizit mit Hilfe einer im Vergleich zur Grundschrift sehr großen Schriftart realisiert werden (etwa `<font size="+4">`). Da derartige Informationen auch durch Cascading Style Sheets maskiert sein können, müssen diese ebenfalls mit in die Analyse einfließen.

3.3 Merkmal-basierte Modulanalyse und Informationsextraktion

Hypertextsorten bestehen, wie in Rehm (in diesem Band) dargestellt, aus obligatorischen sowie optionalen Modulen, die unterschiedlich komplexe Informationsbausteine darstellen. Die derzeit in der Implementierungsphase befindliche Komponente zur Strukturanalyse beliebiger (X)HTML-Dokumente mit Hilfe eines makrostrukturelle und logische Einheiten auszeichnenden Tagsets, das mittels eines eigenen Namespace in die DOM-Repräsentation eingebettet wird, stellt die Vorstufe zur automatischen Ermittlung von Hypertextsortenmodulen in Dokumenten dar. Sobald eine vollständige strukturelle Analyse eines Dokuments vorliegt, dessen Hypertextsorte ebenfalls ermittelt wurde, können mit Hilfe spezieller Regelsysteme den Makrostrukturen diejenigen obligatorischen sowie optionalen Module zugewiesen werden, die laut der formalen Spezifikation einer Hypertextsorte vorliegen können bzw. sollten. Diese Regelsysteme werden in Zukunft auf der Erkennungsseite als eine Ontologie von RDF-Beschreibungen (Lassila/Swick 1999) formalisiert, die jeweils Bedingungen unterschiedlichster Art für das Vorhandensein eines Hypertextsortenmoduls bzw. seiner konstituierenden Merkmale enthalten (vgl. Abb. 1).

Ein atomares Modul, das u.a. in der Hypertextsorte *persönliche Homepage eines Wissenschaftlers* optional enthalten sein darf, ist die *explizite Begrüßung*.[10] Mit Hilfe der Angaben, die von Attributen des `hypnotic`-Namespace gemacht werden, kann eine Regel erstellt werden, die die Existenz dieses Moduls ermittelt. Das Modul liegt vor, falls eine isolierte Zeichenkette (a) im Verhältnis zum Rest eines Dokuments relativ kurz ist, (b) eine spezielle typographische Auszeichnung besitzt, (c) ein Dokument einleitet und (d) eine Begrüßungsfloskel enthält. Neben einer Aufstellung üblicher Begrüßungen wird zur Detektion dieses Moduls insbesondere die bereits angesprochene Analyse relativer Schriftgrößen benötigt, da derartige Begrüßungen auf Webseiten oftmals speziell hervorgehoben werden (Rehm 2003a, 2002a).

10 Beispielsweise „Herzlich Willkommen“ oder „Welcome“, vgl. u.a. de Saint-Georges (1998), Rehm (2002a, 2003a).

Umfangreichere Module können teilweise bereits durch eine Analyse ihrer rein strukturellen Eigenschaften detektiert werden. Das Modul *interessante Links* beispielsweise ist definiert als listenartige Präsentation von mindestens zwei externen Hyperlinks, die evtl. flankiert sind von kurzen Erläuterungen der jeweiligen Informationsangebote, wobei mindestens ein Hyperlink in einem logischen Abschnitt der Liste vorkommt. Eine derartige Definition lässt sich unmittelbar in einen XPath-Ausdruck überführen, der dieses Modul instanziiert, vgl. Listing 1. In XPath ist es möglich, mehrere Ausdrücke – die dann Prädikate genannt werden – durch eckige Klammern zu kombinieren. Der Ausdruck in Listing 1 testet für den Kontextknoten zunächst, ob es sich generell um eine Liste handelt, was in der Analyse-DTD durch das Attribut-Wert-Paar `hypnotic:TagGroup="list"` ausgedrückt wird. Falls diese Liste mehr als ein *list item* enthält und die Anzahl der enthaltenen Hyperlinks größer oder gleich der Anzahl der *list items* ist, wird der Kontextknoten als Modul *interessante Links* analysiert. Der boolesche Ausdruck wird jedoch nur dann wahr, wenn auch das letzte Prädikat erfüllt wird. Dieses besagt, dass, falls in dem Teilbaum beliebige Elemente[11] mit dem Attribut `hypnotic:LinkType` enthalten sind, ausschließlich der Wert `external` erlaubt ist; jeder andere Wert negiert dieses Prädikat und somit den gesamten XPath-Ausdruck. Dies entspricht der Definition dieses Hypertextsortenmoduls, die eingangs paraphrasiert dargestellt wurde.

```
//*[@hypnotic:TagGroup="list"]
   [@hypnotic:NumberOfListItems > 1]
   [@hypnotic:TotalLinkCount >=
    number(@hypnotic:NumberOfListItems)]
   [//*[@hypnotic:LinkType]]
   [(not(.//*[@hypnotic:LinkType and @hypnotic:LinkType !=
    "external"]))]
```

Listing 1: XPath-Ausdruck zur Detektion des Hypertextsortenmoduls „Interessante Links“

Dieser Ausdruck ist in der Lage, in einer Stichprobe von 100 persönlichen Homepages von Wissenschaftlern (siehe Rehm, in diesem Band) die entsprechenden Module zu extrahieren. Bislang bezieht sich die Erkennung jedoch lediglich auf diejenigen Listen, die von den Autoren der Seiten explizit mit den entsprechenden HTML-Elementen (`<ul>`, `<ol>`, `<dl>`) ausgezeichnet wurden. Daher muss in Zukunft eine Erweiterung der Strukturanalyse um Funktionen erfolgen, die auch alternative Formen der Darstellung von Listen als solche

11 Derzeit wird lediglich das HTML-Element `<a>`, das einen Hyperlink markiert, mit dem Attribut `hypnotic:LinkType` versehen. In Zukunft müssen hier auch Image Maps und evtl. alternative Möglichkeiten der Verlinkung mit in die Verarbeitung einbezogen werden.

erkennen, um diese konzise Art der Informationsextraktion auf strukturell voranalysierten HTML-Dokumenten anwenden zu können (s. Abschnitt 3.2).

Falls eine Erkennungsregel bzw. eine Gruppe von Regeln einen Treffer findet, wird der korrespondierende Teilbaum der DOM-Repräsentation als konsumiert markiert, wobei spezielle XML-Elemente das Vorkommen einer expliziten Begrüßung oder einer Liste interessanter Links einbetten. Diese XML-Elemente beziehen sich nun nicht mehr auf die Analyse-DTD, sondern auf die Formalisierung einer Hypertextsorte mit Hilfe von XML Schema (vgl. Rehm, in diesem Band). Generell ist die Konvertierung eines HTML-Dokuments nach XML geplant als eine mit Hilfe der Analyse-DTD durchgeführte massive Anreicherung mit Strukturwissen, das im Zuge der Entdeckung zulässiger Hypertextsortenmodule schrittweise reduziert und durch eine XML-Repräsentation gemäß der Hypertextsorte ersetzt wird, so dass eine explizit strukturierte Hypertextsorteninstanz das Resultat der Verarbeitung darstellt. Abschließend kann die Validierung einer solchen maschinell erstellten Hypertextsorteninstanz gegen die korrespondierende XML Schema Beschreibung erfolgen, um fehlerhaft durchgeführte Auszeichnungen aufdecken zu können (vgl. Rehm 1999).

4 Zusammenfassung und Ausblick

Dieser Beitrag stellt ein Konzept zur automatischen Klassifikation von HTML-Dateien einer eng begrenzten Domäne vor, für die ein Korpus von ca. 4 Mio. Dokumenten aufgebaut wurde. Die beiden generellen Ziele des Vorhabens sind die Schaffung präziserer Möglichkeiten der Informationsrecherche, d.h. die Unterstützung klassischer Information Retrieval-Verfahren mit einer Filter-Komponente, die auf der Spezifizierung der vom Benutzer gewünschten Hypertextsorten beruht, sowie die Realisierung eines abstrakten Ansatzes zur Informationsextraktion, der über die üblichen Wrapping-Verfahren hinausgeht und ebenfalls Struktur- und Inhaltswissen generischer Hypertextsortenbeschreibungen mit einbezieht. Dieses Wissen wird repräsentiert als eine Sammlung von Ontologien, die organisatorische Strukturen, thematische Gruppen sowie die eigentlichen konstituierenden Bausteine – Hypertextsorten-Module – spezifizieren (vgl. Rehm, in diesem Band).

Neben der Implementierung noch ausstehender Analyse-, Klassifikations- und Extraktionsmodule sind zusätzlich zahlreiche weitere empirische Analysen notwendig, um die beteiligten Ontologien in einer Weise zu erweitern, dass sie produktiv eingesetzt werden können. Ebenfalls ist die Integration einer der zahlreichen, im Quelltext verfügbaren Suchmaschinen zur Realisierung des Hypnotic-Prototypen geplant. Offen sind derzeit noch Fragen der Skalierung des dargestellten Ansatzes. Dies betrifft etwa die Adaption an andere Hypertextsorten-Systeme, andere Sprachen oder auch alternative universitäre oder institutionelle Strukturen.

5 Literatur

Asirvatham, Arul Prakash und Kranthi Kumar Ravi: Web Page Classification Based on Document Structure. Technischer Bericht. International Institute of Information Technology, Hyderabad 2001. [http://www.iiit.net/stud_pub.htm]

Bayerl, Petra Saskia: Linguistische Analyse studentischer persönlicher Homepages. Magisterarbeit im Studiengang Germanistik, Institut für deutsche Sprache und mittelalterliche Literatur, Justus-Liebig-Universität, Gießen 2002.

Bray, Tim, Dave Hollander und Andrew Layman: Namespaces in XML. Technische Spezifikation. World Wide Web Consortium 1999.

Carchiolo, Vincenza, Alessandro Longheu und Michele Malgeri: Extracting Logical Schema from the Web. In: *Tan, Ah-Hwee und Philip S. Yu* (Hg.): Proceedings of the International Workshop on Text and Web Mining, Melbourne 2000, S. 64-71.

Chan, Michael und Gin Yu: Extracting Web Design Knowledge: The Web De-Compiler. In: IEEE International Conference on Multimedia Computing and Systems (ICMCS 1999), Bd. 2. Florence: IEEE Computer Society 1999, S. 547-552.

Chen, Jinlin, Baoyao Zhou, Jin Shi, Hongjiang Zhang und Qui Fengwu: Function-Based Object Model Towards Website Adaption. In: Proceedings of the 10th International World Wide Web Conference (WWW-10), Hong Kong 2001, S. 587-596.

Clark, James und Steve DeRose: XML Path Language (XPath). Technische Spezifikation. World Wide Web Consortium 1999.

Cowie, Jim, Evgeny Ludovik und Ron Zacharski: An Autonomous, Web-based, Multilingual Corpus Collection Tool. In: Proceedings of the International Conference on Natural Language Processing and Industrial Applications, Moncton 1998, S. 142-148. [http://crl.nmsu.edu/~raz/langrec/nlpia.htm]

Cowie, Jim und Yorick Wilks: Information Extraction. In: *Dale, Robert, Hermann Moisl und Harold Somers* (Hg.): Handbook of Natural Language Processing. New York, Basel: Marcel Dekker 2000, S. 241-260.

Crowston, Kevin und Marie Williams: Reproduced and Emergent Genres of Communication on the World Wide Web. In: The Information Society 16 (3), 2000, S. 201-215.

de Saint-Georges, Ingrid: Click Here if You Want to Know Who I Am. Deixis in Personal Homepages. In: Proceedings of the 31st Hawaii International Conference on Systems Sciences (HICSS-31), IEEE 1998.

Dillon, Andrew und Barbara A. Gushrowski: Genres and the Web: Is the Personal Home Page the First Uniquely Digital Genre? In: Journal of the American Society for Information Science 51 (2), 2000, S. 202-205.

DiPasquo, Dan: Using HTML Formatting to Aid in Natural Language Processing on the World Wide Web. Senior Honors Thesis, School of Computer Science, Carnegie Mellon University 1998. [http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/]

DuBois, Paul: MySQL. Indianapolis: New Riders 1999.

Eikvil, Line: Information Extraction from World Wide Web – A Survey. Technischer Bericht 945. Norweigan Computing Center 1999.

Fielding, R., J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach und T. Berners-Lee: Hypertext Transfer Protocol – HTTP/1.1. Network Working Group – Request for Comments (RFC) 2616. 1999. [http://www.ietf.org/rfc/]

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

Hawking, David, Nick Craswell und Donna Harman: Results and Challenges in Web Search Evaluation. In: *Tang, E.* (Hg.): The 8th International World Wide Web Conference, International World Wide Web Conference Committee, Foretec Seminars, National Research Council, Canada, Toronto 1999. [http://www8.org/w8-papers/2c-search-discover/results/results.html]

Hors, Arnaud Le, Philippe Hégaret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion und Steve Byrne: Document Object Model (DOM) Level 2 Core Specification. Technische Spezifikation. World Wide Web Consortium 2000.

Hsu, Chun-Nan und Ming-Tzung Dung: Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web. In: Information Systems 23 (8), 1998, S. 521-538.

Koehler, Wallace: Web Page Change and Persistence – A Four-Year Longitudinal Study. In: Journal of the American Society for Information Science and Technology 53 (2), 2002, S. 162-171.

Langer, Stefan: Sprachen auf dem WWW. In: *Lobin, Henning* (Hg.): Sprach- und Texttechnologie in digitalen Medien – Proceedings der Frühjahrstagung der Gesellschaft für Linguistische Datenverarbeitung, Justus-Liebig-Universität Gießen. Gießen: Gesellschaft für linguistische Datenverarbeitung 2001, S. 85-91.

Lassila, Ora und Ralph R. Swick: Resource Description Framework (RDF). Model and Syntax Specification. Technische Spezifikation. World Wide Web Consortium 1999. [http://www.w3.org/TR/REC-rdf-syntax/]

Liu, Ling, Calton Pu und Wei Han: An XML-Enabled Wrapper Construction System for Web Information Sources. In: Proceedings of the International Conference on Data Engineering (ICDE) 2000, S. 611-621.

Müller, Martin: Inducing Conceptual User Models". In: *Wrobel, Stefan* (Hg.): Proceedings of ABIS-99 (Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen). Magdeburg: Gesellschaft für Informatik 1999.
[http://www-mmt.inf.tu-dresden.de/joerding/abis99/proceedings.html]

Myllymaki, Jussi: Effective Web Data Extraction with Standard XML Technologies. In: Proceedings of the 10th International World Wide Web Conference (WWW-10). Hong Kong 2001, S. 689-696.

Pemberton, Steven: XHTML 1.0: The Extensible Hypertext Markup Language (Second Edition). Technische Spezifikation. World Wide Web Consortium 2002. [http://www.w3.org/TR/xhtml1/]

Rehm, Georg: Automatische Textannotation – Ein SGML- und DSSSL-basierter Ansatz zur angewandten Textlinguistik. In: *Lobin, Henning* (Hg.): Text im digitalen Medium – Linguistische Aspekte von Textdesign, Texttechnologie und Hypertext Engineering. Wiesbaden: Westdeutscher Verlag 1999, S. 179-195.

Rehm, Georg: korpus.html – Zur Sammlung, Datenbank-basierten Erfassung, Annotation und Auswertung von HTML-Dokumenten. In: *Lobin, Henning* (Hg.): Sprach- und Texttechnologie in digitalen Medien – Proceedings der Frühjahrstagung der Gesellschaft für Linguistische Datenverarbeitung, Justus-Liebig-Universität Gießen. Gießen: Gesellschaft für linguistische Datenverarbeitung 2001, S. 93-103.

Rehm, Georg: E-Mail-ähnliche Textstrukturen in studentischen Homepages. Unveröffentlichtes Manuskript, Ausarbeitung eines Vortrags auf dem Germanistentag 2001, 30.09.-03.10.2001, Erlangen 2002a.

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

Rehm, Georg: Schriftliche Mündlichkeit in der Sprache des World Wide Web. In: *Ziegler, Arne und Dürscheid, Christa* (Hg.): Kommunikationsform E-Mail. Tübingen: Stauffenburg 2003a, S. 263-308.

Rehm, Georg: Texttechnologie und das World Wide Web – Anwendungen und Perspektiven. In: *Lobin, Henning und Lemnitzer, Lothar* (Hg.): Texttechnologie – Anwendungen und Perspektiven. Tübingen: Stauffenburg 2003b. [Erscheint.]

Roussinov, Dmitri, Kevin Crowston, Mike Nilan, Barbara Kwasnik, Jin Cai und Xiaoyong Liu: Genre Based Navigation on the Web. In: Proceedings of the 34th Hawaii International Conference on Systems Sciences (HICSS-34), IEEE 2001.

Sahuguet, Arnaud und Fabien Azavant: Building Intelligent Web Applications Using Lightweight Wrappers. In: Data and Knowledge Engineering 36 (3), 2001, S. 283-316.

Shepherd, Michael und Carolyn Watters: The Functionality Attribute of Cybergenres. In: Proceedings of the 32nd Hawaii International Conference on Systems Sciences (HICSS-32), IEEE 1999.

Shin, Christian, David Doermann und Azriel Rosenfeld: Classification of Document Pages Using Structure-Based Features. In: International Journal of Document Analysis and Recognition 3, 2001, S. 232-247.

Walker, Derek: Taking Snapshots of the Web with a TEI Camera. In: Computers and the Humanities (33), 1999, S. 185-192.