XDOC – XML-basierte Werkzeuge für multilinguale Korpora

Dietmar RÖSNER

Zusammenfassung

Im Projekt XDOC werden Konzepte und XML-basierte Werkzeuge für den Umgang mit alltäglichen Dokumenten entwickelt. Im Projekt werden derzeit unterschiedliche Korpora bearbeitet. Neben Texten aus medizinischen Lehrbüchern und multilingualen technischen Dokumentationen gehört dazu insbesondere ein Korpus mit elektronischer Post.

1. Einführung

Eine zentrale Fragestellung für die Forschungen in der AG Wissenbasierte Systeme und Dokumentverarbeitung ist die nach dem Zusammenhang zwischen Dokumenten und Wissen. Bei den bisherigen Arbeiten stand die Generierung von Dokumenten aus repräsentiertem Wissen im Vordergrund (RÖSNER 1994). Dabei wurden insbesondere multilinguale technische Dokumente bearbeitet und den Fragen der Diskursstruktur und der Autorenunterstützung wurde besondere Aufmerksamkeit zuteil (RÖSNER et al. 1997). In den aktuellen Arbeiten wird nun versucht, die bei der Generierung gewonnenen Erfahrungen auch für die komplementäre Aufgabe nutzbar zu machen: Wie lässt sich das in Dokumenten enthaltene Wissen erschließen und extrahieren?

Das Material für diese Arbeiten sind wieder alltägliche Gebrauchstexte.

Die Dokumente haben zwar unterschiedliche Funktionen und auch die möglichen Anwendungen sind für die einzelnen Textsorten unterschiedlich, aber viele Teilaufgaben bei der Dokumentanalyse stellen sich unabhängig von der Dokumentverwendung für jedes Dokument.

Dies motiviert die im folgenden vorgestellten Arbeiten an einem ‘Werkzeugkasten’ zur Dokumentanalyse.

2. Beispielanwendung: Verarbeitung von emails

Elektronische Post liefert nicht nur ein gutes Beispiel für den Bedarf und die Nützlichkeit multilingualer Werkzeuge für den Umgang mit großen Dokumentbeständen. Bei email ist die Verarbeitung gegenüber dem allgemeinen Fall auch etwas erleichtert: Email-Dokumente liegen bereits in elektronischer Form vor. Sie müssen nicht erst durch Scannen und OCR und eine mögliche nachfolgende Fehlerbehandlung verarbeitbar gemacht werden. Emails sind des weiteren semistrukturierte Dokumente, einige der Informationen aus dem Header können für die Weiterverarbeitung sinnvoll herangezogen werden.

Auf der anderen Seite erfordert die Verarbeitung von Emails aber Lösungen für viele der Teilaufgaben, die auch für den Umgang mit anderen alltäglichen Dokumenten (z.B. Geschäftsbriefen) erforderlich sind:

- So muss das zumindest rudimentär vorhandene Layout interpretiert werden, um relevante Strukturierungen zu erkennen (z.B. Anreden, Hervorhebungen, Tabellenstrukturen, Zitationen anderer emails, usw.).
- Neben Standardklassen für lexikalische Kategorien (Nomen, Verben, Adjektive etc.) spielen nicht-lexikalische Kategorien eine große Rolle (z.B. Abkürzungen, email-Adressen, URLs, Datumsangaben) und müssen sicher detektiert und weiterverarbeitet werden können.
- Es muß davon ausgegangen werden, daß die Lexika für die offenen Wortklassen und Kategorien wie Orts- und Personennamen nie vollständig sein werden und daher oft die robuste Kategorisierung unbekannter Terme in den Texten erforderlich ist.

3. Warum XML?

Hier kann und soll keine technische Einführung in XML gegeben werden. Für die weiteren Ausführungen soll folgende Skizze der Motivation für XML (BOSAK 1996) und der daraus abgeleiteten technischen Festlegungen ausreichen:

Die Arbeitsgruppe, die im Auftrag des WWW-Konsortiums die Version 1.0 des XML-Vorschlags entwickelt hat, wollte aus den Erfahrungen mit HTML und SGML die erforderlichen Konsequenzen ziehen (XML 1998).

Ohne HTML hätte sich das WWW nicht so rasant verbreitet, aber HTML hat für viele Anwendungen gravierende Beschränkungen:

- HTML ist zu sehr an Fragen der Präsentation orientiert und bietet zu wenig Möglichkeiten, die Struktur und Semantik von Informationen festzulegen,
- die Menge der verfügbaren Tags ist vorgegeben und kann vom Anwender nicht einfach für seine Dokumente erweitert werden,
- Verarbeitung von Daten in HTML-Seiten erfolgt in erster Linie beim Server.

Mit SGML kann die logische Struktur und die Semantik von Dokumenten flexibel erfasst werden. Der schiere Umfang des Standards und viele Elemente, die eher selten in Anwendungen genutzt, für normkonforme Implementationen aber gefordert wurden, haben die Akzeptanz von SGML sehr erschwert.

XML, die Extensible Markup Language, bietet einerseits das, was HTML fehlt, also die Möglichkeit, eigene Mengen von Tags und eigene Informationsstrukturen zu definieren und einen großen Teil der Verarbeitung auf die Clients zu verlagern. Andererseits wurde für die manchmal auch als '*SGML on the Web*' apostrophierte Sprache XML auf viele Elemente von SGML verzichtet, die als nicht essentiell eingestuft wurden. So fallen z.B. die sog. Minimierungen weg, mit denen zwar der Speicheraufwand um den für die weggelassenen Zeichen verringert, aber das Parsen von ausgezeichneten Dokumenten unnötig kompliziert gemacht wurde.

XML kann verschiedene Aufgaben erfüllen: Bei der ursprünglichen Konzeption stand die Strukturierung von Dokumenten im Vordergrund; die Charakterisierung der logischen Struktur für eine Klasse von Dokumenten ist die Aufgabe der Dokumenttypdefinition (DTD). XML erlaubt aber auch, Metadaten zu kodieren. Diese können sich auf das Dokument als ganzes oder auf einzelne Teile beziehen. Werden die Regeln für wohlgeformte Dokumente beachtet, so kann auch ganz auf eine DTD verzichtet werden und Auszeichnungen können an beliebige Dokumentelemente vergeben werden. Im Prinzip könnte dies bis zur Ebene einzelner Zeichen erfolgen. In XDOC ist es zum Beispiel sinnvoll (s.u.), einzelne Wörter mit Tags für ihre Wortklasse zu versehen. Diese Tags können als Attribute weitere Angaben zum Verarbeitungsprozeß enthalten.

4. Komponenten des XDOC-Systems

Für unsere Arbeiten haben wir den eigenen Email-Verkehr herangezogen. Im Material tauchen insbesondere die Sprachen Englisch und Deutsch auf, gelegentlich Französisch. Eine erste Erkennungsaufgabe ist, automatisch die jeweils vorherrschende Sprache einer Email zu bestimmen. Dies ist erforderlich, damit die sprachspezifischen der nachfolgend angewendeten Werkzeuge korrekt aktiviert werden können.

Für Englisch werden verschiedene Werkzeugkästen zur Dokumentverarbeitung für Forschungszwecke zur freien Verfügung gestellt. Dazu gehören die Systeme GATE (GATE 1988) und LT XML (LTG 1999). Mit diesen experimentieren wir mit den englischen Dokumenten aus unseren Korpora.

Vergleichbare Systeme für Deutsch stehen derzeit nicht zur Verfügung. Wir haben uns in unseren eigenen Arbeiten daher auf Deutsch beschränkt und entwickeln solche Komponenten, die wir nicht oder nicht in der für unsere Zwecke benötigten Funktionalität von anderen Forschungsgruppen übernehmen oder anpassen können.

Zu den in XDOC derzeit realisierten weiteren Basisdiensten für die Dokumentverarbeitung gehören:

- Strukturerkennung und -auszeichnung,
- Taggen von Interpunktion,
- Taggen der Wortklassen,
- n-Gramm-Analyse,
- robustes partielles Parsen.

Diese Basisdienste illustrieren wir im folgenden, zum Teil auch mit Beispielen aus dem Korpus.

Identifikation der Sprache: Die vorherrschende Sprache eines Dokuments wird nach dem relativen Anteil einschlägiger Stopwörter (aus geschlossenen Wortklassen) an der gesamten Zahl von Wortkandidaten festgelegt. Wörter aus offenen Wortklassen werden dazu nicht herangezogen. Damit ist das Ergebnis dieser Komponente nicht von der jeweiligen Abdeckung des Lexikons für offene Wortklassen abhängig.

```
IMAIL(190): (guess-primary-language "Hallo,
fuer das Treffen heute um 15:00 Uhr: Der Raum D304m ist belegt
(FR-Sitzung). Wir treffen uns daher erst einmal bei mir in G206.
Gruss,
Detlef Nauck
")
"GERMAN"
IMAIL(191): (guess-primary-language "Hallo,
this is just to remind you of our meeting.
Gregor
")
"ENGLISH"
```

Strukturerkennung und -auszeichnung:

Die generelle Aufgabe ist hier, Blöcke zusammengehörenden Fließtexts von anderen Elementen des Dokuments zu separieren. Bei emails gehört dazu u.a., den Nachspann mit meist Adressdaten abzutrennen und zitierte emails zu detektieren.

Taggen von Interpunktion:

In diesem Verarbeitungsschritt wird versucht, die Interpunktion in den identifizierten Fließtexten korrekt zu interpretieren und so Sätze und andere linguistische Einheiten für die Weiterverarbeitung zu identifizieren. Zu den dabei auftretenden Problemen gehört, daß Interpunktionszeichen auch für andere Zwecke verwendet werden können. Ein Punkt kann z.B. in einer Abkürzung, einer Datumsangabe, einer email-Adresse, in Ordinal- oder Gleitkommazahlen und einigen anderen Umgebungen verwendet werden.

Einige der Schwierigkeiten bei der korrekten Erkennung von Interpunktion zeigt das folgende Beispiel:

```
IMAIL(196): (setf text
"Tagging realistischer Texte sollte auch Abkuerzungen (wie z.B. PD
oder Abt.) und andere nichtlexikalische Elemente - man denke an
Email-Adressen wie roesner@iws.cs.uni-magdeburg.de, Zahlen oder
Datumsangaben - beruecksichtigen.")


IMAIL(197): (tag-ip-line text)
"Tagging realistischer Texte sollte auch Abkuerzungen <IP>(</IP>wie
<ABBR>z.B.</ABBR> PD oder <ABBR>Abt.</ABBR><IP>)</IP> und andere
nichtlexikalische Elemente - man denke an Email-Adressen wie
roesner@iws.cs.uni-magdeburg.de<IP>,</IP> Zahlen oder
Datumsangaben - beruecksichtigen<IP>.</IP>"
```

Taggen der Wortklassen:

Für die englischen Texte wird der Brill-Tagger eingesetzt. Für das Taggen deutschsprachiger Texte wird eine eigene Entwicklung auf der Basis der Morphologiekomponente MORPHIX (FINKLER and NEUMANN 1988) verwendet. Da MORPHIX die flektierten Formen der geschlossenen Wortklassen robust erkennen kann, aber keinerlei Kontext einbezieht und so z.B. Artikel und Relativpronomen nicht unterscheidet, dienen die eigenen Erweiterungen vor allem dazu, einerseits isolierte Mehrfachanalysen durch Bezug zum Kontext zu disambiguieren und andererseits völlig unbekannte Wörter zu klassifizieren. Hierfür wird mit verschiedenen Herangehensweisen experimentiert. Dazu gehört die Anwendung von Heuristiken, die sich entweder auf die unbekannte Wortform (z.B. Endungen) selbst oder auf die relative Stellung zu anderen Wörtern oder auch Interpunktionszeichen beziehen können. Eine andere Technik ist (s.u.), beim Parsen Wörter mit unbekannter Wortklasse zuzulassen und aus erfolgreicher Anwendung von Grammatikregeln Kategorisierungshypothesen (für offene Wortklassen) abzuleiten.

Für den Beispieltext von oben ergibt sich folgendes Ergebnis, wenn zunächst die Interpunktion und dann die Wortklassen bestimmt und getaggt werden:

```
IMAIL(198): (tag-wordclasses-in-line (tag-ip-line text))

"<XXX>Tagging</XXX> <XXX>realistischer</XXX> <N>Texte</N>
<V>sollte</V> <ADV>auch</ADV> <N SRC = GH1>Abkuerzungen</N>
<IP>(</IP><MULT VAL = (\"FRAGEADVERB\"\"S-KONJ\")>wie</MULT>
<ABBR>z.B.</ABBR> <ABBR>PD</ABBR> <K-KONJ>oder</K-KONJ>
<ABBR>Abt.</ABBR><IP>)</IP> <K-KONJ>und</K-KONJ> <ADJ>andere</ADJ>
<XXX>nichtlexikalische</XXX> <N>Elemente</N> <IP>-</IP>
<PARTIKEL>man</PARTIKEL> <V>denke</V>
<MULT VAL = (\"PRP\"\"VZ\")>an</MULT> <N SRC = GH1>Email-Adressen</N>
<MULT VAL = (\"FRAGEADVERB\"\"S-KONJ\")>wie</MULT>
<E-ADR>roesner@iws.cs.uni-magdeburg.de</E-ADR><IP>,</IP>
<MULT VAL = (\"N\"\"V\")>Zahlen</MULT> <K-KONJ>oder</K-KONJ>
<N SRC = GH1>Datumsangaben</N> <IP>-</IP> <XXX>beruecksichtigen</XXX>
<IP>.</IP>
"
```

Die für die lexikalischen Klassen verwendeten Tags (z.B. `<N>`, `<V>`, ...) und die für die nichtlexikalischen (z.B. `<E-ADR>` für Email-Addressen) sollten selbsterklärend sein. `<XXX>` kennzeichnet unklassifizierte lexikalische Objekte. `<MULT ...>` wird für lexikalische Objekte vergeben, für

die ohne Berücksichtigung des Kontexts mehrere Lesarten existieren. Diese werden dann in einer Liste als Wert des Attributs VAL aufgeführt. Wird eine Wortklasse aufgrund einer Heuristik bestimmt, so wird als Prozessinformation der Name der verwendeten Heuristik als Wert des Attributs SRC angegeben (wie in <N SRC = GH1>Abkuerzungen</N>).

Die Wirkung von Heuristiken zur Auflösung von Ambiguitäten wird mit dem folgenden Beispiel verdeutlicht. Das Resultat, das beim Tagging erzielt wird, bei dem nur isolierte lexikalische Elemente berücksichtigt werden, enthält Mehrfachkategorisierungen:

```
IMAIL(80): (tag-text "Der Mann gab die Tat zu.")
"<MULT VAL = (\"DETD\"\"RELPRO\")>Der</MULT> <N>Mann</N> <V>gab</V>
<MULT VAL = (\"DETD\"\"RELPRO\")>die</MULT> <V>Tat</V> <MULT VAL =
(\"PRP\"\"S-KONJ\"\"VZ\")>zu</MULT><IP>.</IP>"
```

Durch die Anwendung von Heuristiken lassen sich hier alle, im allgemeinen Fall zumindest einige der Mehrdeutigkeiten auflösen:

```
IMAIL(81): (improve-tagger-result *)
"<DETD SRC = MH1>Der</DETD> <N>Mann</N> <V>gab</V>
<V>gab</V> <DETD SRC = MH1>die</DETD> <V>Tat</V>
<VZ SRC = MH2>zu</VZ><IP>.</IP>"
```

Die Heuristiken für die Entscheidung zwischen alternativen Wortklassen (wie zwischen definiter Artikel oder Relativpronomen für ‘der’ bzw. ‘die’ oder zwischen Präposition, subordinierender Konjunktion oder Verbzusatz für ‘zu’) berücksichtigen den unmittelbaren Kontext. Oft wird dabei negativ entschieden: Liegt z.B. bei der Alternative definiter Artikel vs. Relativpronomen kein unmittelbar vorausgehendes Komma oder keine unmittelbar vorausgehende koordinierende Konjunktion vor, so wird definiter Artikel gewählt (ein vorhandenes Komma allein führt aber zu keiner Entscheidung bei dieser Alternative).

Alle verfügbaren Heuristiken werden wiederholt auf einen Text anzuwenden versucht. Dies ist erforderlich, da durch Disambiguierungen in vorausgegangenen Schritten manchmal erst die Voraussetzung zur Anwendung einer Heuristik geschaffen wird. Da angewandte Heuristiken stets zu einer Reduktion der Zahl der offenen Alternativen führen, endet diese Iteration nach endlich vielen Schritten.

Da nur unmittelbarer Kontext einfließt und keine komplexen linguistischen Analysen bei der Entscheidung über Alternativen zur Verfügung stehen, müssen Heuristiken dieser Art sehr sorgfältig formuliert werden. Auch ist stets zu prüfen, ob implizite Annahmen über den vorherrschenden Stil in einem bearbeiteten Korpus in die formulierten Heuristiken einfließen und diese daher nicht für alle möglichen, sondern nur für die in einer Subsprache typischen Wortklassenmuster gelten.

So fließt bei der oben beschriebenen Heuristik zur Wahl zwischen definitem Artikel oder Relativpronomen die Annahme ein, daß Strukturen der folgenden Art nicht erwartet werden: „Der dich angerufen hat, der Mann, hat seinen Namen nicht genannt.".

n-Gramm-Analyse

Für die Entwicklung subsprachenspezifischer (Partial-)Grammatiken haben sich Werkzeuge als nützlich erwiesen, mit denen Vorkommenshäufigkeiten für aus Tags gebildete n-Gramme bestimmt und statistisch ausgewertet werden können. So lassen sich die in einem Korpus besonders häufigen Konstruktionen detektieren. Ergänzt durch den schnellen Zugriff auf Fundstellen für solche Strukturen im Korpus ergibt sich eine wertvolle Hilfe für den Grammatikentwickler.

Robustes partielles Parsen

Für unsere Anwendungen ist insbesondere das robuste Erkennen von komplexen Nominalphrasen wichtig. Hierzu wird ein Chart-Parser eingesetzt, der auch bei unvollständig getaggtem Material robust arbeitet und Kategorisierungshypothesen bildet.

Die Arbeitsweise des robusten Parser wird durch die folgenden Ausschnitte aus einem Ablaufprotokoll und anhand der internen Ergebnisstrukturen verdeutlicht:

```
IMAIL(94): (improve-tagger-result
(tag-text "das unbekannte System"))
"<DETD SRC = MH1>das</DETD> <XXX>unbekannte</XXX> <N>System</N>
"
IMAIL(95): (chart-parse-sentence (improve-tagger-result
(tag-text "das unbekannte System")))
...
"unbekannte"assumed to belong to wordclass ADJ
```

```
"unbekannte"assumed to belong to wordclass ADV
"unbekannte"assumed to belong to wordclass ADJ
"unbekannte"assumed to belong to wordclass V
"unbekannte"assumed to belong to wordclass V
...
total result(s) :
(0 3 (NP :CAS MORPHIX:AKK :NUM MORPHIX:SG :GEN MORPHIX:NTR)
((DETD "das") (XXX "unbekannte":AS ADJ) (N "System")) NIL)
(0 3 (NP :CAS MORPHIX:NOM :NUM MORPHIX:SG :GEN MORPHIX:NTR)
((DETD "das") (XXX "unbekannte":AS ADJ) (N "System")) NIL)
```

Wenn eine Grammatikregel ein lexikalisches Element aus einer der offenen Wortklassen benötigt, an der aktuellen Position aber ein mit `<XXX>` getaggtes Element vorliegt, wird für dieses – wenn auch weitere Bedingungen wie Groß-/Kleinschreibung erfüllt sind – mit der Annahme weitergearbeitet, daß das nicht klassifizierte Element zu der gesuchten Wortklasse gehört. Die Annahme wird festgehalten (wie in `(XXX "unbekannte":AS ADJ)`). Wie man an den Systemmeldungen sieht, ist es zwar von der Organisation der Grammatik abhängig, aber bei einem solchen Ansatz unvermeidlich, daß während der Verarbeitung zunächst auch falsche Hypothesen gebildet werden. Für diese wirken dann aber Grammatikregeln, die größere Bereiche der Eingabe zu überspannen versuchen, als effektive Filter.

Der Parser verwendet – wie oben ersichtlich – intern (für die Darstellung und Verarbeitung der Grammatikregeln und der Chart) zwar LISP-Datenstrukturen. Für die weitere Nutzung werden Analyseergebnisse aber wieder in XML-Strukturen umgesetzt.

Interoperabilität

Zu den Anforderungen an einen ‘Werkzeugkasten’ – nicht nur einen zur Dokumentanalyse – gehören:

- die einzelnen ‘Werkzeuge’ (Module) sollen flexibel und vielseitig einsetzbar sein,
- die ‘Werkzeuge’ sollen einerseits generell sein, sich aber auch spezialisieren lassen.

Die konsequente Verwendung von XML hilft, diese Forderungen zu erfüllen. Alle Komponenten des XDOC-Systems operieren auf im XML-Stil ausgezeichneten Texten (Dateien oder Textstrings) und liefern ihre Ergebnisse im gleichen Formalismus. Damit ist es leichter möglich, Komponenten flexibel zu kombinieren und wie beim ‘piping’ in UNIX komplexe Funktio-

nalität durch Hintereinanderausführen elementarer Komponenten zu realisieren. Ein vergleichbarer Ansatz hat sich auch im Projekt LT XML der Language Technology Group in Edinburgh mit Werkzeugen für englische Korpora bewährt.

5. Aktuelle Arbeiten

Die Basisdienste von XDOC werden in einer Reihe von laufenden Experimenten erprobt und, wo erforderlich, weiterentwickelt.

Bei der Anwendung email-Verarbeitung wird versucht, emails zu klassifizieren, um sie dem Nutzer entsprechend seinen Präferenzen vorzusortieren. Zunächst wurde hierzu nur Information aus dem Header herangezogen. Bei den aktuellen Arbeiten wird nun damit experimentiert, Information auch aus dem Inhalt zu extrahieren (so z.B. thematische Schwerpunkte und Daten aus Tagungsankündigungen).

Parallel dazu laufen Experimente zur Konzept- und Terminologieextraktion auf einem Korpus mit medizinischen Lehrbuchtexten und einem mit multilingualen technischen Dokumentationstexten.

Literatur

BOSAK, Jon (1996): XML, JAVA, and the future of the web. `http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm`.

FINKLER, W. and NEUMANN, G. (1988): MORPHIX. A fast Realization of a classification-based Approach to Morphology. In: TROST, H. (ed.): *4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop – Wissensbasierte Sprachverarbeitung*, 11-19, Berlin etc.: Springer.

LTG (1999): Language Technology Group, LT XML version 1.1. `http://www.ltg.ed.ac.uk/software/xml/` .

GATE (1988): H. CUNNINGHAM, Y. WILKS, R. GAIZAUSKAS, GATE – a general architecture for text engineering. In: *Proceedings of COLING-96*.

RÖSNER et al. (1997): D. R., B. GROTE, K. HARTMANN and B. HÖFLING, From Natural Language Documents to Sharable Product Knowledge: A Knowledge Engineering Approach. In: *Journal of Universal Computer Science (JUCS)* 3(8), 955-987.

RÖSNER (1994): Automatische Generierung von mehrsprachigen Instruktionstexten aus einer Wissensbasis. Habilitationsschrift, Fakultät Informatik der Universität Stuttgart.

XML (1998): Tim BRAY, Jean PAOLI, C.M. SPERBERG-MCQUEEN, Extensible Markup Language (XML) 1.0. `http://www.w3.org/TR/1998/REC-xml-19980210` .