Exposé einer Diplomarbeit
Entwicklung einer Benutzerschnittstelle für die Suche in linguistischen mehrebenen Korpora unter Betrachtung softwareergonomischer Gesichtspunkte.

Karsten Hütter
huetter@informatik.hu-berlin.de

3. April 2008

1 Einführung

Die Suche in annotierten Textdaten ist ein wichtiges Hilfsmittel in der modernen Sprachwissenschaft, von dem mehr und mehr Linguisten Gebrauch machen. Die Vorteile bei der systematischen, computergestützten Suche in Korpusdaten liegen vor allem in der Belegsuche und statistischen Analysen über annotierten Textdaten. Dabei kann es sich um verschiedenartige geschriebene Texte oder um transkribierte gesprochene Sprache handeln.

Die Annotationen gehören meist nicht zu den Primärdaten und werden nachträglich automatisch, halbautomatisch oder manuell hinzugefügt. Sehr häufig werden Wortart (Part of Speech) und Wortstamm (Lemma) mit Hilfe von spezialisierter Tagger-Software annotiert (vgl. Schmid, 1994, für die automatische Annotation von Wortarten). Zusätzlich werden abhängig von den Forschungsfragen, die mit dem Korpus bearbeitet werden sollen, spezielle Annotationen erstellt. Hierbei kann es sich bspw. um Syntaxbäume über den Textdaten, um Annotationen der Informationsstruktur oder im Fall von Lernerkorpora um die Markierung und Klassifizierung von sprachlichen Fehlern handeln (Siemen et al., 2006).

Zur Illustration unterschiedlicher Annotationsebenen Abbildung 1 entnommen aus Lezius (2002b).

```
Ein Mann kommt, der lacht.
```

Die Annotationsebene direkt unter dem Ausgangssatz zeigt die automatisch erstellte Ebene der Wortarten. Darunter ist eine genauere Klassifikation der Token nach Genus, Kasus und Numerus zu sehen (morphologische Annotation). Über dem Text befindet sich die grafische Repräsentation der syntaktischen Struktur des Satzes. Gerade bei manuellen Annotationen wird deutlich, dass es sich immer um eine Interpretation des Ausgangstextes handelt. Die Einflüsse dieses Interpretationsprozesses müssen später bei der Nutzung berücksichtigt werden. Hierbei handelt es sich aber um ein prinzipielles Problem, auf das im Folgenden nicht weiter eingegangen werden kann.

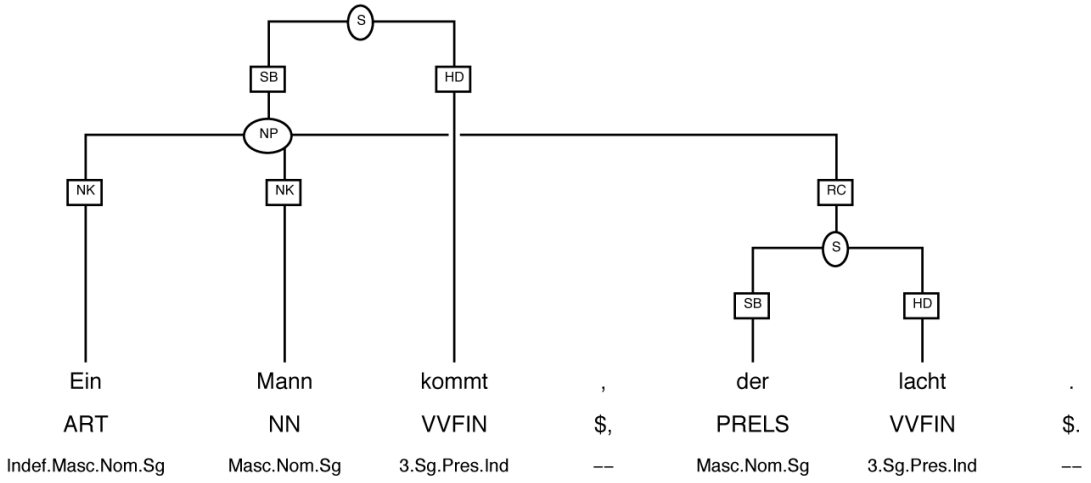

Abbildung 1: Annotierter Beispielsatz aus dem Tiger-Korpus.

Korpusdaten sind durch ihre Struktur und Menge nicht immer einfach zu handhaben. Deswegen werden seit Jahren Suchsysteme für spezielle Daten entwickelt. Im englischsprachigen Raum sind die bekanntesten Systeme die *Penn Treebank* mit dem Suchinterface *CorpusSearch* (Marcus et al., 1994, http://www.cis.upenn.edu/~treebank/) und das *British National Corpus* mit dem Suchinterface *Xaira* (Burnard, 2000, http://www.natcorp.ox.ac.uk/), für den deutschsprachigen Raum ist es das *TigerKorpus* mit *TIGERSearch* (Lezius, 2002a, http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/). *Xaira* und *TigerSearch* mit dem Zusatzwerkzeug *TigerRegistry* können darüber hinaus vielfältige andere XML-Korpusdaten durchsuchen. Leider besitzt fast jedes Korpus eine eigene Suchplattform/Bentzerschnittstelle mit individuell sehr unterschiedlichen Methodologien. Die Suche über mehrere verschiedenartige Korpora ist daher auch dann nicht möglich, wenn sie vergleichbare Annotationen (Annotationsebenen- und kategorien) besitzen.

Im Rahmen des Sonderforschungsbereiches SFB 632 "Informationsstruktur: Die sprachlichen Mittel der Gliederung von Äußerung, Satz und Text"[1] soll im Teilprojekt D1 "Datenbank für Informationsstruktur: Annotation und Retrieval"[2] eine flexible Suchplattform für Mehrebenenkorpora geschaffen und einer breiten Nutzergruppe zur Verfügung gestellt werden. Dies ist eine große Herausforderung sowohl für die Entwicklung der eigentlichen Suche als auch für die Erarbeitung einer allgemeinen Benutzerschnittstelle.

[1] http://www.sfb632.uni-potsdam.de/main3322.html
[2] http://www.sfb632.uni-potsdam.de/~d1/

1.1 Aktueller Stand des Teilprojektes D1

Der erste Schritt für die Umsetzung eines solchen Suchsystems war die Entwicklung der Abfragesprache *ANNIS Query Language ANNISQL* und die Entwicklung des darauf aufbauenden webbasierten Suchsystems *ANNIS* (Götze und Dipper, 2006, http://www.sfb632.uni-potsdam.de/~d1/annis/). *ANNISQL* ist syntaktisch an die Baumstruktur linguistischer Daten angelehnt. Sie beschreibt die gesuchten Teilgraphen durch die Eigenschaften der Knoten und Kanten. Deshalb bietet *ANNISQL* Vorteile in der Vermittlung für Neueinsteiger und ist für diverse Mehrebenenkorpora gut geeignet.

Die permanente Speicherung der Daten erfolgt im flexiblen XML-standoff-Format *Paula* (Dipper (2005), Wörner et al. (2006)). Die *Paula*-Daten werden in eine Hauptspeicherrepräsentation des Graphen überführt und können dann durchsucht werden. Die stärkste Einschränkung dieser Vorgehensweise ist die Platzbegrenzung des Hauptspeichers. Insbesondere kann der Java Heap auf 32bit-Systemen eine Gesamtgröße von 1.996.800kb (≈2Gb) nicht überschreiten. Damit ist diese Herangehensweise für große Korpora ungeeignet.

Deswegen soll zukünftig die von Faulstich et al. (2006) vorgestellte Anfragesprache zur Suche verwendet werden. *DDDQuery* beinhaltet ein Datenbankmodell zur Ablage von Korpora und wird mit Hilfe eines Compilers in SQL-Anfragen übersetzt. *DDDQuery* ist eine sehr umfangreiche und damit komplexe Abfragesprache, mit der die meisten Endbenutzer nicht konfrontiert werden können. Daher soll zukünftig *ANNISQL* nach *DDDQuery* überführt werden.

Auch die Umsetzung der *ANNIS*-Benutzerschnittstelle bietet aufgrund mangelnder Übersichtlichkeit Ansatzpunkte für Erweiterungen und Verbesserungen. Die Anforderungen dafür sind aktuell nur unvollständig ausgearbeitet.

2 Ziel

Ziel der Diplomarbeit ist es, eine neue webbasierte Benutzerschnittstelle für *ANNISQL* über Mehrebenenkorpora zu entwickeln. Die wichtigste Anforderung an dieses Interface sind eine flexible, verständliche Darstellung der Korpusdaten bzw. Suchergebnisse, ein Werkzeug zur grafischen Erstellung von Anfragen, die Unterstützung von statistischen Anfragen und der Export der Suchergebnisse in einem vielseitig verwendbaren Format. Hierfür sind z.B. CSV[3] für die Tabellenansichten und *Paula* für die Baumstrukturen denkbar.

Das Hauptaugenmerk soll aber nicht ausschließlich auf den Features der Anwendung liegen. Es ist weitgehend bekannt, dass Nutzerfreundlichkeit ein wichtiges Qualitätskriterium von Hard- und Software ist (Mayhew und Bias, 1994). Um eine langfristig nachhaltige

[3]CSV: Comma Separated Values. Ein Austauschformat für tabellarische Daten.

Anwendung zu entwickeln, muss deshalb ingenieurspsychologischen Aspekten besondere Beachtung zukommen.

3 Vorgehen

Zuerst findet eine Auswahl von Usability-Kriterien für die Benutzerschnittstelle statt. Hierbei steht die DIN EN ISO 9241 als Leitnorm und Basis vieler Normen der Softwareergonomie im Mittelpunkt - insbesondere deren "Grundlagen für die Dialoggestaltung" (Teil 10) (Beimel et al., 1994). In Abgrenzung dazu kann auf Grund der hohen Anforderungen an die Interaktivität des Systems eine Barrierefreiheit im Sinne der Norm ISO/TS 16701 nicht gewährleistet werden. Die Lauffähigkeit auf den gängigen Browsern kann jedoch sichergestellt werden.

Um die vorhandenen Anforderungen bestätigen und klassifizieren zu können, wird ein Online-Fragebogen zur genauen Erfassung der positiven und negativen Nutzerkritiken zu einer Auswahl verfügbarer Systeme erstellt. Aufgrund des engen Zeitrahmens kann dieser nur oberflächliche Kriterien abfragen und von einer kleinen Gruppe von Benutzern im Rahmen des Sonderforschungsbereichs beantwortet werden. Genaue Aussagen über die Nutzerzufriedenheit und resultierende Anforderungen lassen sich damit zwar nicht treffen, aber sicherlich zeigt sie den Handlungsbedarf und gibt neue Impulse für das Projekt.

Die Entwicklung der Software erfolgt als Prototyping. Es werden frühzeitig Experten und fachkundige Anwender in den Entwicklungsprozess einbezogen um die Qualität und die Akzeptanz der fertigen Software sicherzustellen. Durch die rasche Einbindung des Prototypen in reale Arbeitsprozesse können Konzeptions- und Implementierungsfehler früh erkannt, diskutiert und behoben werden. Nicht zuletzt durch die Verwendung dieser Methode ist sichergestellt, dass die Software für reale Arbeitsprozesse eingesetzt wird.

Als praktisches Werkzeug für die Gestaltung der Interaktion zwischen Nutzer und System werden für das Projekt spezifische Personas eingeführt. Personas dienen als Modelle realer Nutzergruppen. Sie stellen u.a. den fachlichen Wissensstand, die Computer Literacy[4] und Gebrauchsmotive realer Nutzer dar. Das von Cooper (1999) vorgestellte Persona-Konzept ist auch heute noch ein fester Bestandteil des Usability Engineerings.

Das Webinterface wird auf dem JavaScript-Framework EXTJS[5] entwickelt. EXTJS bietet alle notwendigen Funktionen zur Gestaltung Fenster-orientierter Anwendungen wie es die Designstudie in Abbildung 2 zeigt. Dadurch ist es für den Benutzer möglich, erlernte Umgangsweisen aus grafischen Betriebssystemen auf eine Webapplikation zu übertragen, so dass Vertrautheit geschaffen und die Lernschwelle gesenkt wird. Mit der Verwendung

[4]Computer Literacy beschreibt das Wissen und die Fähigkeit, Technologie im Allgemeinen und Computer im Speziellen effizient nutzen zu können.

[5]EXTJS Website: http://extjs.com

asynchroner Kommunikation zwischen Webbrowser und Webserver mittels AJAX[6] kann dem Benutzer stets direkte Rückmeldung auf dessen Eingaben und ausgeführte Aktionen gegeben werden. Die umfangreichen Mechanismen von EXTJS für die Ein- und Ausgabe sind die Grundlage für kurze Entwicklungszyklen und damit ideal für das Prototyping.

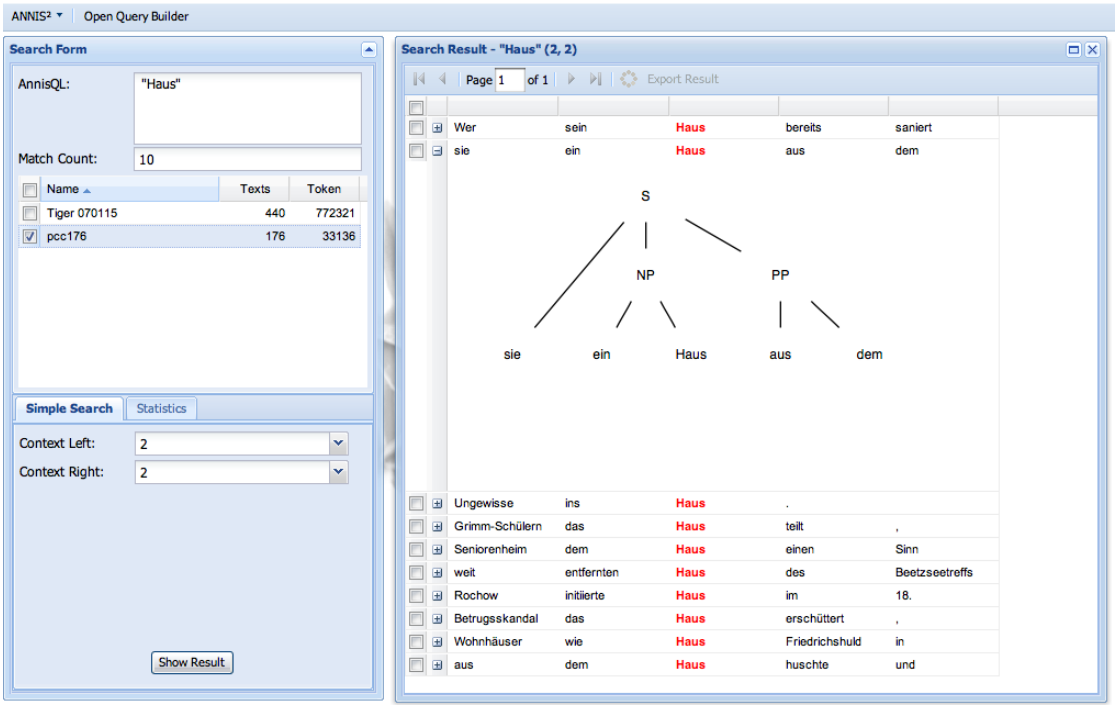

Abbildung 2: Designstudie der fensterorientierte Suchoberfläche.

Die Kommunikation mit dem Suchbackend findet über einen Java RMI Service statt. Dieser bietet der Webapplikation alle notwendigen Funktionen zur Abfrage von verfügbaren Korpora und deren Metadaten, von Suchergebnissen und statistischen Auswertungen. Dieser Service übernimmt die Umwandlung der *ANNISQL*-Anfrage in *DDDQuery* und führt diese dann im Backend aus. Für eine einzelne Suchanfrage ergibt sich dann der in Abbildung 3 dargestellte Informationsfluss.

[6] AJAX: "Asynchronous JavaScript and XML" steht für das Konzept des asynchronen Datenaustausches zwischen Webbrowser und Webserver und bricht mit und stellt eine Ergänzung zum Request-Response-Paradigma des WWW dar.

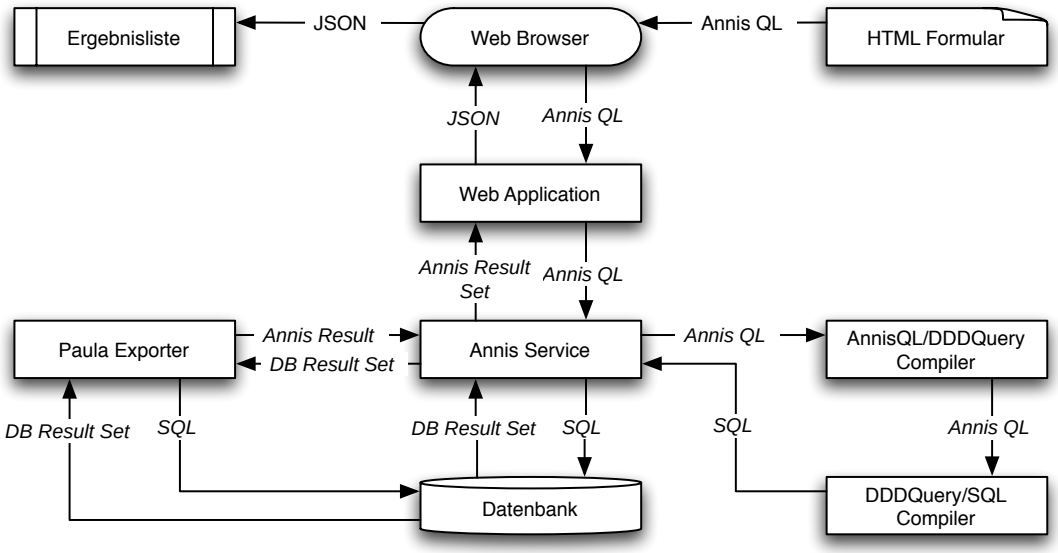

Abbildung 3: Informationsfluss vom HTML-Suchformular bis zur HTML-Ausgabe der Ergebnisliste.

Die logische Aufteilung des Gesamtsystems bietet den Vorteil, dass sowohl das Frontend als auch das Backend zu einem späteren Zeitpunkt ausgetauscht werden können, ohne sich zu beeinflussen. Hier ist insbesondere die Anbindung neuer Suchbackends durch eine andere Implementierung des Service oder die Entwicklung neuer Client-Anwendungen unter Nutzung des Service denkbar. Auch die parallele Benutzung einer Serviceinstanz mit unterschiedlichen Client Applikationen ist möglich.

Damit eine reibungslose Kommunikation zwischen Frontend und Suchbackend erfolgen kann, muss die Schnittstelle genau definiert werden. Hierfür wurde ein Java-Paket mit Interfaces für den Service und die zurückgegebenen Datenobjekte erstellt.

DDDQuery wird derzeit aufbauend auf Vitt (2005) einschließlich einiger Optimierungen fertiggestellt. Noch ist es nicht möglich, alle für das Frontend notwendigen Informationen zu liefern. Außerdem ist die aktuell importierte Datenbasis noch sehr klein und nicht repräsentativ. Um aber das Prototyping wie geplant durchführen zu können wird zumindest ein funktionaler Service benötigt, der *ANNISQL*-Anfragen entgegennehmen und die Schnittstelle mit repräsentativen Rückgabewerten bedienen kann.

Literatur

Jurgen Beimel, Raimund Schindler, und Hartmut Wandke. Do Human Factors Experts Accept the ISO 9241 Part 10 – Dialogue Principle – Standard? *Behaviour and Information Technology*, 13(4):299–308, 1994.

Lou Burnard. Reference Guide for the British National Corpus (World Edition). *Oxford Universtity Computing Services*, 2000.

Alan Cooper. The Inmates are Running the Asylum. In *Software-Ergonomie*, 1999.

Stefanie Dipper. XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In *Berliner XML Tage 2005 (BXML 2005)*, Seiten 39–50, Berlin, Germany, 2005.

Lukas C. Faulstich, Ulf Leser, und Thorsten Vitt. *Current Trends in Database Technology – EDBT 2006*, Ausgabe Volume 4254/2006, Kapitel Implementing a Linguistic Query Language for Historic Texts., Seiten 601–612. Springer Berlin / Heidelberg, 2006. URL `http://www.deutschdiachrondigital.de/publikationen/qlqp.pdf`.

Michael Götze und Stefanie Dipper. ANNIS: Complex Multilevel Annotations in a Linguistic Database, 2006. URL `http://acl.ldc.upenn.edu/W/W06/W06-2709.pdf`.

Wolfgang Lezius. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*, Ausgabe 8. AIMS, 2002a.

Wolfgang Lezius. Tiger Search - Ein Suchwerkzeug für Baumbanken. Vortrag für Konvens, 2002b.

Mitchell P. Marcus, Beatrice Santorini, und Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313, 1994.

Deborah J. Mayhew und Randolph G. Bias, editors. *Cost-Justifying Usability*. Morgan Kaufmann, 1994.

Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, 09 1994. URL `http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf`.

Peter Siemen, Anke Lüdeling, und Frank Henrik Müller. FALKO - Ein fehlerannotiertes Lernerkorpus des Deutschen. Vortrag für Konvens, 2006.

Thorsten Vitt. DDDquery: Anfragen an komplexe Korpora. Diplomarbeit, Humboldt-Universität zu Berlin, Institut für Informatik, Berlin., 2005. URL `http://www.deutschdiachrondigital.de/publikationen/dddq.pdf`.

Kai Wörner, Andreas Witt, Georg Rehm, und Stefanie Dipper. Modelling Linguistic Data Structures. In *Proceedings of the Extreme Markup Languages 2006*, Montréal, Canada, 2006.