Komponenten einer modularen Softwarearchitektur zur Entwicklung und Anwendung der Methoden der Korpusanalyse

Andreas Blätte, Stand 1. Mai 2013 (Entwurf)

1. Ausgangslage

Die Geistes- und Gesellschaftswissenschaften stehen durch die Digitalisierung in technischer Hinsicht vor zwei Aufgaben und Herausforderungen. Erstens stellt sich die Herausforderung der Schaffung digitaler Datensammlungen. Dabei geht es sowohl um eine aufholende Erschließung bereits bestehender Materialbestände, die es bislang nicht in einem für die maschinelle Verarbeitung geeigneten Datenformat gibt, als auch um die Schaffung von Verfahren, um die Entwicklungen der durch Digitalisierung geprägten Gesellschaft fortlaufend zu erfassen. Insofern es sich dabei um Text handelt, sind Archive und flüchtiges digitales Material als Korpora aufzubereiten. Das digitale Material ist allerdings für sich genommen bedeutungslos, wenn nicht Möglichkeiten gefunden werden, dieses in der Forschung produktiv zu nutzen. Daher gilt es – zweitens - eine digitale Infrastruktur zu schaffen, die es sowohl in der Forschung als auch in der Lehre nicht nur einem kleinen, esoterischen Kreis möglich macht, mit neuen digitalen Materialbestände zu arbeiten.

Software für die Text- bzw. Korpusanalyse gibt es selbstverständlich längst. Es gibt einige Programme für Verfahren der sozialwissenschaftlichen computerunterstützten Inhaltsanalyse (z.B. Wordstat) und solche, die ihren Ursprung in der Tradition von Korpusanalyse (bzw. der französischen Lexikometrie) haben (WordSmith, Lexico3). Die genannten Programme sind für die Installation auf einem einzelnen Arbeitsplatzrechner (Laptop etc.) gedacht[1] und sie bieten dem Nutzer mit einer graphischen Nutzeroberfläche einigen Komfort. Genau dies ist zugleich der Nachteil dieser Software: Es sind geschlossene, nur durch den jeweiligen Softwareanbieter erweiterbare Systeme. Dies schränkt die Möglichkeiten ein, neue oder für bestimmte Daten spezialisierte Verfahren der Korpusanalyse zu entwickeln.

Wie kann eine Softwarearchitekur beschaffen sein, die bessere Entwicklungsperspektiven eröffnet? Es sollen in einem ersten Schritt die Kriterien näher bestimmt werden, denen eine solche Softwarearchitektur genügen sollte. In einem zweiten Abschnitt wird ein Überblick über ein Modell gegeben, das für die Datenhaltung die Corpus Workbench (CWB) nutzt, für textstatistische Analysen und für die Methodenentwicklung das Statistikpaket R einsetzt und das in verschiedener Weise mit grafischen Benutzeroberflächen versehen werden kann. Die einzelnen Komponenten werden in Abschnitt 3 erläutert und diskutiert. Am Schluss wird diskutiert, welchen möglichen Einschränkungen die Entwicklungsperspektiven eine solche Architektur unterliegt.

[1] Vgl. hierzu die Unterscheidung verschiedener Generationen der Software bei Hardie (im Erscheinen).

2. Kriterien

Die folgenden Kriterien erscheinen bei der Einschätzung einer Softwarearchitektur relevant, die möglichst gute Entwicklungsperspektiven für die Korpusanalyse in den Geists- und Gesellschaftswissenschaften bietet.

Open Source: Schwellen zur Nutzung von Korpora werden gesenkt, wenn die Komponenten einer Softwarearchitektur für die Korpusanalyse unter allgemeinen öffentlichen Lizenzen zur Verfügung stehen. Studierende könnten Korpusanalyse meiden, wenn diese die Anschaffung teurer Lizenzen erfordert. Es hemmt den wissenschaftlichen Austausch, wenn man sich über die Implementation von Auswertungsverfahren nicht umfassend austauschen kann.

Dokumentation: Die Implementierung korpusanalytischer Auswertungsverfahren erfordert Programmierarbeit. Ad hoc entwickelter oder schlecht dokumentierter Code sperrt sich gegen eine Wiederverwendung und führt dazu, dass man für spezifische Forschungsprobleme immer wieder neuen Code schreibt, obwohl alte Funktionen hätten wiederverwenden können. Dies behindert die Möglichkeiten, Fortschritte zu erzielen. Code sollte im Sinne des Ideals eines kumulativen Wissensfortschritts so dokumentiert sein, dass dieser geteilt und wiederverwendet werden kann.

Modularität: Neue Auswertungsverfahren sollten modular eingeführt werden können. Auch grafische Benutzerschnittstellen sollten nach Bedarf eingefügt werden können. Auf diese Weise können etablierte Verfahren implementiert werden und neue nach und nach entwickelt werden.

Portabilität: Die Architektur sollte verschiedenen Nutzerszenarien entsprechend installiert werden können. Die Implementierung korpusanalytischer Verfahren ist oft rechenintensiv. Dies spricht dafür, speicher- und rechenintensive Operationen auf einem leistungsfähigen Server durchzuführen. Zuweilen will man aber auch offline mit Korpora arbeiten können. Oder man möchte sich bei der Entwicklung eines neuen Verfahrens die Mühe ersparen, Skripte ständige zwischen Rechnern hin- und herzuschieben. Es sollte auch Windows-Usern möglich sein, dass System zu nutzen. Die gesamte Architektur sollte leicht portierbar sein.

Schutz von Urheberrechten: Während die Komponenten einer Softwarearchitektur allgemein verfügbar sein sollte, können Korpora nicht immer vollumfänglich zugänglich gemacht werden. Etwa dann, wenn lizenzrechtlich geschützte Presseerzeugnisse als Korpus genutzt werden, sollen oder dürfen Nutzer keinen unmittelbaren Zugriff auf die vollen zugrundeliegenden Korpora haben. In einem solchen Fall sollten Nutzer zwar die Möglichkeit der textstatistischen Auswertungen, aber keinen Durchgriff auf die Datenschicht haben.

Benutzerfreundlichkeit: Korpusanalyse ist technisch nicht trivial. Selbst wenn es eine ideale Infrastruktur gäbe, setzt methodisch reflektierte Forschung mit Korpora voraus, dass man das Zustandekommen, die Eigenschaften und Eigenheiten der untersuchten Korpora kennt und dass man die Verfahren der Korpusanalyse versteht. Trotzdem sollte das Ziel sein, dass Korpusanalyse so wenig „Voodoo“ wie möglich ist. Die wissenschaftliche Legitimität korpusanalytischer Methoden profitiert, wenn korpusanalytische Methoden leicht zugänglich und benutzerfreundlich anwendbar sind. Einstiegshürden sollten nicht hoch sein, die Nutzung von Korpora in politikwissenschaftlichen BA- oder MA-Kursen sollte nicht praktisch unmöglich sein. „Usability“ ist aber auch für denjenigen von

Interesse, die als Fortgeschrittene oder als Forscher korpusanalytisch arbeiten. Auch sie wollen sich das Leben nicht unnötig schwer machen.

3. Vorschlag einer Softwarearchitektur

Hier soll eine Architektur mit den folgenden drei „Schichten“ vorgeschlagen werden, die den genannten Kriterien entsprechen kann:

- Corpus Workbench (CWB) als System zur Korpusverwaltung (Datenschicht)

- R als Statistikpaket zur Analyse der Texte (Anwendungsschicht)

- Verschiedene Web-Applikationen (Präsentationsschicht)

Das Schaubild im Anhang zeigt, wie diese Komponenten aufeinander aufbauen.

3.1 Datenschicht: Nutzung der Corpus Workbench (CWB)

Die vorgeschlagene Architektur soll in der Lage sein, einen effizienten Zugriff auch auf größere, linguistisch annotierte Korpora zu ermöglichen. Eine Indizierung des Korpus ist eine Voraussetzung dafür. Die Corpus Workbench (CWB)[2] ist ein etabliertes System für die Korpusverwaltung, das sich dafür anbietet. Die CWB stellt insbesondere mit der Abfragesprache CQP (Corpus Query Processor) Möglichkeiten einer Suche nach komplexen sprachlichen Ausdrücken zur Verfügung. Die CWB ist gut dokumentiert, es stehen Einführungen und Tutorials zur Verfügung[3], die die Vermittlung von CQP-Kenntnissen im Lehrkontext erleichtern. Die CWB ist unter einer öffentlichen Lizenz (GNU General Public License) verfügbar.

Die CWB erfordert ein spezifisches Importformat, bei dem das Korpus in eine vertikalisierte Form gebracht wird. Eine XML-Annotation des Korpus kann verarbeitet werden. Wenn für den Zugriff auf ein mit der CWB verwaltetes Korpus CQPweb verwendet werden soll, sind allerdings die Anforderungen für die „strukturelle“ Annotation des Korpus sehr spezifisch. Die dafür erforderlichen Datentransformationen sind normalerweise technisch kein Problem. Probleme, die das aufwirft, werden unten diskutiert.

3.2 Anwendungsschicht: Datenanalyse mit R

Für die Analyse von Korpora sind Perl, Python, R und Java etablierte Skriptsprachen. Perl bietet sich durch eine nahtlose Einbindung regulärer Ausdrücke an und wurde früh von Korpuslinguisten genutzt. Python-Code ist gegenüber Perl leichter lesbar. Python wird wohl in der Computerlinguistik oft eingesetzt. Es stehen für die Arbeit mit Korpora spezialisierte Bibliotheken (z.B. NLTK[4]) zur Verfügung, für die auch gute Einführungen zur Verfügung stehen (…). Sowohl für Perl als auch für

[2] http://cwb.sourceforge.net/
[3] http://www.bubenhofer.com/korpuslinguistik/kurs/index.php?id=cwb_start.html
[4] http://nltk.org/

Python gibt es APIs, welche die Nutzung der CWB für die Korpusverwaltung ermöglichen. Dies gilt auch für Java. Gegenüber Perl und Python ist Java zwar bei Computerlinguisten weit verbreitet, doch ist die Einarbeitung für Sozialwissenschaftler mit einem höheren Aufwand verbunden.

Die folgenden Erwägungen sprechen für die Verwendung von R. Für R gibt es mit rcqp ein Paket zum Zugriff auf die CWB.[5] Perl und Python haben unbestreitbare Vorteile bei der Aufbereitung von Korpora. Bei der Datenauswertung ist es jedoch ein großer Vorteil, wenn eine Skriptsprache standardmäßig einen interaktiven Modus zur Verfügung stellt, was bei Perl nicht der Fall ist. Gegenüber Python hat R durch eine konsequente Ausrichtung auf die statistische Datenanalyse Vorteile. Es steht eine Vielzahl statistischer Methoden schon standardmäßig zur Verfügung. R bietet exzellent entwickelte Verfahren für die Visiualisierung von Ergebnissen. Damit soll nicht gesagt sein, dass es entsprechende Module (NumPy, SciPy etc.) bei Python nicht gäbe – die Integration der Statistik ist bei R nur nahtloser.

R zwingt bei der Entwicklung von Packages zur konsequenten, für andere Nutzer zugänglichen Dokumentation von Funktionen. Dies erhöht die Wahrscheinlichkeit des Austauschs von Routinen. Vor allem ist R in den Sozialwissenschaften als (kostenfreie) *open source*-Alternative zu kommerziellen Statistikpaketen (wie SPSS, STATA) weithin akzeptiert. Für viele Sozialwissenschaftler ist die Hemmschwelle hoch, eine „Programmiersprache“ wie Python zu erlernen. R ist auch nur eine Skriptsprache – aber man kennt diese bereits und viele haben zumindest gehört, dass R weitreichende Möglichkeiten bietet.

Für die Analyse linguistischer Daten mit R gibt es bereits mehrere Lehrbücher.[6] Es gibt eine Reihe von Paketen für die Arbeit mit Korpora. Das Paket ‚rcqp’ ermöglicht den Zugriff auf die mit der CWB indizierten Korpora, einschließlich der Nutzung von CQP. Mit diesem lassen sich nicht die Tabellen und statistischen Auswertungsverfahren generieren, die für die Korpusanalyse im eigentlichen Sinne erforderlich sind. Dafür sind weitere Funktionen bzw. Pakete erforderlich, die sich aber in R gut implementieren lassen. Für Auswertungen der im PolMine-Projekt entstehenden Korpora wird ein Paket entwickelt, dass für jene spezialisiert ist.

3.3 Präsentationsschicht / Web-Applikationen

CQP ist bei der Nutzung auf der Kommandozeile ein mächtiges Werkzeug, R bietet die ganze Bandbreite statistischer Auswertungsmöglichkeiten. Die Kommandozeile ist allerdings für viele potenzielle Nutzer abschreckend. Daher erscheint es sinnvoll, von Anfang an die Möglichkeit einer Erweiterung um eine grafische Benutzeroberfläche im Blick zu haben. Solche Erweiterungen sollten modular erfolgen können, so dass neu entwickelte Funktionen ohne große Probleme mit einer grafischen Benutzeroberfläche versehen werden können. Dies ist nicht nur für Einsteiger und weniger computeraffine Nutzer interessant. Auch für denjenigen, der ein Verfahren entwickelt hat, ist es oft schlicht angenehmer, wenn man mit einer graphischen Benutzerfläche arbeiten kann.

Bei der vorgeschlagenen Nutzung der CWB und von R gibt es (unter anderem) die folgenden Möglichkeiten, graphische Nutzeroberflächen zu schaffen:

[5] http://www.bubenhofer.com/sprechtakel/2012/06/05/statistische-analysen-von-korpora-mit-r-direkt-auf-die-cwb-zugreifen/

[6] Gries, Baayen

CQPweb: Mit CQPweb steht ein fertig entwickeltes Paket zum Zugriff auf die CWB zur Verfügung. CQPweb erweitert diese um einige Funktionen (Kollokations- und Schlagwortberechnung), so dass grundlegende textstatistische Analysen möglich sind.

RStudio: Mit „RStudio“ kann R lokal mit einer graphischen Benutzeroberfläche/IDE versehen werden. „RStudio Server“ ist die Server-Variante von RStudio.[7] Nutzer finden auf diese Weise ein vollständig konfiguriertes System vor. Die Dokumentationsfunktionen von R sind verfügbar, grafische Auswertungen können vom Nutzer lokal gespeichert werden etc.

Shiny-Web-Applikationen: Vom gleichen Entwicklerteam wie RStudio wird ‚shiny‘ angeboten.[8] Dies ist ein R Paket, das eine schnelle und einfache Entwicklung kleiner Web-Applikationen ermöglicht, die R-Funktionen aufrufen.

PHP-Wrapper: Hat man eine R-Funktion, können schließlich auch schlichte PHP-Wrapper angelegt werden, um eine R-Funktion aufzurufen.

Die genannten Möglichkeiten können sowohl serverseitig, als auch über einen lokalen Webserver genutzt werden. Für Web-Anwendungen und für eine lokale Nutzung muss kein jeweils spezifischer Code generiert werden. Die Bereitstellung einer serverseitigen Umgebung kann allerdings den Vorteil haben, dass man bei Korpora, die man noch nicht vollständig frei weitergeben kann, einen Zugriff auf das Korpus selbst nicht herstellt und doch einem interessierten Nutzerkreis Auswertungsmöglichkeiten erschließen kann. Dies kann etwa ein Modell für Printmedien sein, die man als Korpus aufbereitet hat.

4. Diskussion

Nachteile einer Entscheidung für R sind, dass in R Verfahren nachgebildet werden müssten, die vielleicht anderweitig schon bestehen. R findet unter Umständen nicht bei allen Disziplinen Akzeptanz. Für Computerlinguisten kann die Nutzung von Python oder Java naheliegender sein. Eine Entscheidung für R reduziert unter Umständen die Möglichkeiten eines interdisziplinären Austauschs. Dem gegenüber steht das Potenzial, Sozialwissenschaftler für eine Arbeit mit Korpora zu gewinnen.

R hat längst eine große Entwicklergemeinschaft an sich gebunden. Man kann davon ausgehen, dass R langfristig gepflegt wird. Weniger gesichert ist dies für rcqp als CWB-Schnittstelle. Von grundlegender Bedeutung ist, wie zukunftsträchtig die Entscheidung für die CWB ist. Es stellt sich erstens die Frage nach einer Maximalgröße von CWB-Korpora. Ein Korpusumfang von 2 Milliarden Token ist bislang eine Obergrenze, die von der CWB verarbeitet werden kann. Auch wenn dieses Volumen mit den jetzt zur Verfügung stehenden Korpora noch nicht erreicht wird, ist es nur eine Frage der Zeit, bis man diese Größenordnung erreicht. Das Plenarprotokollkorpus umfasst bereits in der jetzigen Version (2000-2013) rund 330 Millionen Token. Die einzige gegenwärtige Barriere, den zeitlichen Umfang bis zur Gründung der Bundesrepublik zu erweitern, ist die Qualität der OCR-Texterkennung. Verbessern sich diese Möglichkeiten, ist Material zu verarbeiten, dass die genannte Obergrenze deutlich überschreiten dürfte. Ein umfassendes Zeitungskorpus wäre sehr groß. Da allerdings die CWB kontinuierlich gepflegt wird, werden diese Probleme wohl gelöst werden.

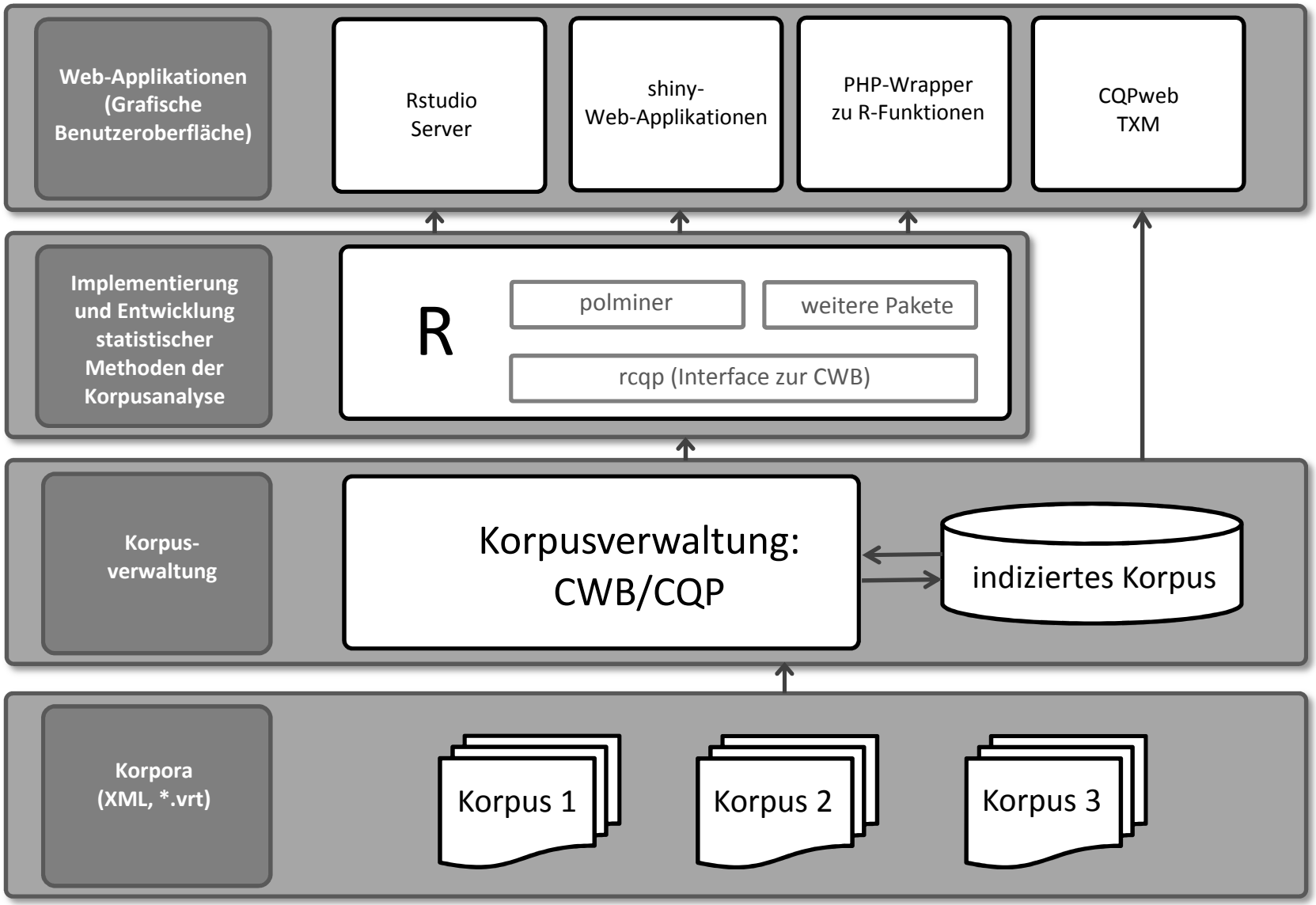
[7] http://www.rstudio.com/ide/download/
[8] http://www.rstudio.com/shiny/

Weniger eindeutig lässt sich die Frage nach der Zukunft von CQPweb beantworten. Vor allem im (politikwissenschaftlichen) Lehrkontext bietet CQPweb gute Möglichkeiten. Ein technischer Vorbehalt gegen CQPweb resultiert – erstens - aus den sehr spezifische Anforderungen, die CQPweb an die Aufbereitung von Korpora stellt. Korpora müssen durch eine Transformation in ein spezifisches Importformat gebracht werden, das von CQPweb verarbeitet werden kann. Konkret ist dies die Anforderung, eine sehr flache XML-Struktur zu generieren, bei der alle Metainformationen zu einem Text, die als Selektionskriterium für die Bildung von Subkorpora herangezogen werden können, Attribute eines „text"-Wurzelknotens/Elements sind. Im Fall der Plenarprotokollkorpora bedeutet dies etwa, jeden einzelnen Abschnitt ununterbrochener Rede zu einem eigenen „text"-Element zuzuordnen und alle Metainformationen einer Debatte hier unterzubringen. Die eigentliche logische Struktur einer Debatte wird damit aufgelöst. Eine offene Frage ist desweiteren, ob CQPweb kontinuierlich weiterentwickelt wird – dies ist noch nicht abzusehen

Gerade hier zeigen sich allerdings die Vorteile der modularen Konzeption der vorgeschlagenen Architektur. So lange die CWB und R als System zur Korpusverwaltung tragfähig bleiben, können relativ flexibel Komponenten der vorgeschlagenen Softwarearchitektur wie CQPweb ersetzt werden. Die Hoffnung bleibt, zu einem Zusammenspiel von Komponenten zu finden, das es erleichtert, die Korpusanalyse in den Geists- und Gesellschaftswissenschaften weiter zu etablieren.

Offene Fragen / Discussion Points:

- Bestimmung des Verhältnisses zu TXM
- Auseinandersetzung mit TextGrid, D-Spin etc.


Web-Applikationen (Grafische Benutzeroberfläche)
Rstudio Server
shiny-Web-Applikationen
PHP-Wrapper zu R-Funktionen
CQPweb TXM
Implementierung und Entwicklung statistischer Methoden der Korpusanalyse
R
polminer
weitere Pakete
rcqp (Interface zur CWB)
Korpus-verwaltung
Korpusverwaltung: CWB/CQP
indiziertes Korpus
Korpora (XML, *.vrt)
Korpus 1
Korpus 2
Korpus 3