

Für die TXSTEP-Version Februar 2017 stehen folgende Dateien und Test-Skripte bereit, die zum größten Teil aus Kursmaterial der Tübinger Tustep-Kurse von W. Ott zusammengestellt wurden:

Dateien (Verzeichnis txstep/uebq)

inh	Inhaltsverzeichnis eines Sammelbandes, Tustep-Datei je 1 Satz pro Beitrag, z.B. Autor, Adam: Titel (1-11)
inh.txt	wie inh, aber Fremd-Datei.
jnh	wie inh, jedoch mit Fehlern: teilweise fehlendes Blank nach Satzzeichen, falsche Reihenfolge der Beiträge.
inh.xml	wie inh, jedoch xml-Datei, jedes Element in eigenem Satz. Tags: <inhalt> <beitrag> <a>... <t>...</t> <s>...</s> </beitrag> ... </inhalt>
inhb5.xml	wie inh.xml, jedoch bei mehrfach-Autoren jeder Autor zwischen <a> und .
atsf	wie inh, Tustep-Datei, jedoch Bestandteile eines Eintrags vorn durch Kennungen =a= =t= =s= markiert, mit vielen Fehlern (doppelte Kennungen, fehlende Kennungen, falsche Reihenfolge der Kennungen, unbekannte Kennungen).
atsf.txt	wie atsf, jedoch Fremd-Datei.
dedale	Tustep-Datei mit französische Wörtern, Reihenfolge der Wörter auf S. 1 entspricht den franz. Sortierregeln (für Sortierübungen; kein vorbereitetes Skript).
span	Tustep-Datei mit spanischen Wörtern, zwei Seiten mit identischem Inhalt, Sortierung in Seite 1 wie für die traditionelle Sortierung des Spanischen (für Sortierübungen; kein vorbereitetes Skript).
sans	Tustep-Datei mit Anfangs-Buchstaben von Sanskrit-Wörtern in der für die Sortierung relevanten Reihenfolge (für Sortierübungen; kein vorbereitetes Skript).
ibam	Schwätzer-Satire des Horaz (sermones I,9; Hexameter, mit Angabe der Versmetrik hinter jedem Vers: 0=spondeus, 1=dactylus; Gedichtnummer=Seitennummer, Versnummer=Zeilennummer).
ibamr	wie ibam, jedoch Gedicht- und Versnummer explizit in [...] vor jedem Vers.

- ibamr.txt wie ibamr, aber Fremd-Datei.
- ibamt.txt wie ibamr.txt, aber ohne Metrik.
- kolinfo Verzeichnis der Tübinger Kolloquien zur EDV in den Geisteswissenschaften (TUSTEP-Datei):
 <ko>1. Kolloquium: 17. 11.1973</ko>
 <au> Vorname <fn>Name</fn> (Institut);
 Vorname <fn>Name</fn>:
 <ti> Thema
 <au> ... (falls mehr als ein Thema)
 <ti> ... " " " " "
- kolinfo.txt wie kolinfo, aber Fremd-Datei.
- lucas Anfang des Lukas-Evangeliums, griechisch, mit Akzentfehler auf dem 1. Wort (Gravis statt Akut).
- lucas.xml wie lucas, aber xml-Datei.
- schella.x 3 Versionen eines Auszugs aus einer Schelling-Edition
 schellb.x (Tustep-Dateien mit xml-Tags).
 schellc.x Die Datei schella enthält den Editionstext / die Kollationsgrundlage, die Dateien schellb und schellc enthalten zu Demo-Zwecken veränderte Versionen des gleichen Textes.
- schella.xml wie schella.x bzw.
 schellb.xml schellb.x bzw.
 schellc.xml schellc.xml, jedoch xml-Dateien.
- schellkac.xml
 (Beschreibung siehe unten beim Skript insert_app.xml)
- bibl.txt Bibliographie 1982: Computer im Editionsweisen
 &.ulVorwort, &.ulTeil1, &.ulTeil2
 dort: @a Autor @t Titel @u unselbst. @s selbst.
- bibl_see.txt wie bibl.txt, aber mit laufend nummerierten Einträgen (z.B. @n 1: @a Autor...) und Verweisen am Eintrags-Ende auf diese Nummern bei Publikationen aus dem gleichen Verlag bzw. der selben Zeitschrift (same publiser / periodical: @m 7 @m 50).
- persons.xml Personenregister (Seck) mit Referenzen
- biographies.xml
 zu den Namen in persons.xml zugehörige Kurz-Infos.
- recker.xml Übungsdatei aus TEI-mail von Ute Recker-Hamm vom 26.4.11;
 Aufgabe: jedes Wort zwischen <w> und </w> setzen.
- gedicht.xml Gedichtsammlung
- gedicht.dtd DTD zu gedicht.xml
- gedichtstransflk.xsl
 xslt-Script, das in ged_xsl.xml mit benutzt wird.

ref_input.xml
Übungsdatei für ref-script.xml und ref.xsl

bau1 der 9. Artikel aus den "12 Artikel der Bauern"
bau2 in 12 verschiedenen Fassungen, z.T. proprietäre Codierung
bau3 der Sonderzeichen.
bau5
bau7
bau33
bau_bau
bau_germ
bau_heli
bau_mem
bau_och
bau_parv

Dateien im Verzeichnis txstep/ueb

txstep.tu Datei für den erzeugten TUSTEP-Code.
Gedicht.dtd wie in /uebq, für die erzeugte XML-Datei.
schellkac.xml
 Kopie von ../uebq/schellkac.xml

Dateien im Verzeichnis txstep/ueb/schell

fra3.html frame für html-Fassung der mit
 schell-html_build.xml erstellten Edition.
texta-u.html Überschrift für Version a (über Textfeld
 in der linken Spalte).
textb-u.html Überschrift für Version b (über Textfeld
 in der rechten Spalte).
textc-u.html Überschrift für Version c (über Textfeld
 in der rechten Spalte).
variantena-u.html Überschrift für Apparat zur Version a (unter
 Textfeld in der linken Spalte).
variantenb-u.html Überschrift für Apparat zur Version b (unter
 Textfeld in der rechten Spalte).
variantenc-u.html Überschrift für Apparat zur Version c (unter
 Textfeld in der rechten Spalte).

TXSTEP-Skripte (Verzeichnis txstep/scripts)1. Datei-Transformation; Text-Analyse

inh2xml.xml	Tustep-Datei inh in eine xml-Datei umwandeln nach dem Muster der Datei inh.xml (s.o.).
inh_f2xml.xml	dto. für inh.txt (= inh im "Fremddaten-Format").
inh.xsl	dto. als xslt-Script.
inh2xml_2.xml	wie inh2xml.xml, andere Lösung.
inh_f2xml_2.xml	dto. für inh.txt (= inh im "Fremddaten-Format").
inh2xml_2n.xml	wie inh2xml_2.xml, aber Autoren nummerieren.
inh2xml_a.xml	wie inh2xml.xml, dabei gleichzeitig Autor, Adam in Adam AUTOR umwandeln.
inh_lnr.xml	Datei inh.xml: vor den Autornamen dreistellige laufende Nummer einfügen.
xml2inh.xml	Datei inh.xml: Tags entfernen, je Eintrag eine Zeile nach dem Muster der Datei inh (s.o.).
inh_length.xml	Datei inh.txt: Länge der einzelnen Beiträge errechnen und einsetzen, Länge des kürzesten und des längsten am Schluss ausgeben (<at-end-of-data><pass>...).
inh_length_2.xml	wie inh_length.xml, aber (<insert-at-end>...).
inh_length_3.xml	wie inh_length_2.xml, aber <brackets><opening-bracket> statt <read-values-from>.
inh_length_3b.xml	wie inh_length_3.xml, aber zusätzlich <closing-bracket>.
inh_length_4.xml	wie inh_length.xml, aber nur wenn nach : "information" oder nach ; "autor"; im Titel "e" vor "information" zu "E" austauschen.
inh_length_mrk.xml	wie inh_length.xml, zusätzlich den kürzesten und den längsten entsprechend markieren.
inh_length_proz.xml	Datei inh.txt: Länge der einzelnen Beiträge und Anteil (in Prozent) an der Gesamtlänge errechnen und einsetzen, Gesamt-Seitenzahl ausgeben
inh_title_length.xml	In Datei inh.xml Zahl der Wörter im Titel errechnen und als Attribut zu <t words="n"> einsetzen; Schlussmeldung: "<summary>There are n words in n titles = n.n words/title</summary>".
names_pages.xml	aus Datei inh eine xml-Datei erstellen, in der nur die Namen der Autoren und die Seitenzahlen genannt sind, nach dem Muster:

- `<name>Hindle, Anthony</name> <name>Raper, Diane</name>
<pages>27-54</pages>.`
- names_pages2.xml wie names_pages.xml, andere Lösung (AKn/EKn statt nur XX).
- names_pages3.xml wie names_pages2.xml, andere Lösung
((Kn/)Kn statt AKn/EKn).
- names_pages3b.xml wie names_pages3.xml ((Kn/)Kn mit eliminate statt select).
- names_pages4.xml wie names_pages3.xml, aber ein Eintrag je Autor
`<name>Hindle, Anthony</name><pages>27-54</pages>
<name>Raper, Diane</name><pages>27-54</pages>`
- name_pages.xml aus Datei inh eine xml-Datei erstellen, in der nur die Namen der Autoren und die Seitenzahlen genannt sind, ein Element je Autor, nach dem Muster
`<item><name><familyName>Holm</familyName>,
<givenName>Bart E.</givenName></name>
<pages>5-26</pages>
</item>`
- name_pages_2.xml wie name_pages.xml, andere Lösung (ETE statt XX in Schleife für jeden Autor).
- vor_famnam.xml Kopie der Datei inh erstellen, in der die Vornamen der Autoren vor den in Versalien umgewandelten Familiennamen stehen.
- aut_ref.xml Verweise vom zweiten und weiteren Autoren (TA+, 2 Dg.) auf den ersten Autor erzeugen, z.B.
Verfasser, Zeno see Autor, Adam
- aut_ref_2.xml wie aut_ref.xml, andere (kürzere) Lösung (KEN, KSP).
- aut_ref_3.xml wie aut_ref.xml, dritte Lösung (ZFZ/ZSP statt ZF+ in Schleife).
- snr_rest.xml Kopie der Datei inh erstellen, in der die Seitenangaben vor den Autorennamen stehen.
- snr_rest2.xml wie snr_rest.xml, andere Lösung (subunits statt text-parts);
Titel nicht mit ausgeben.
- word_search.xml Aus Datei inh die Beiträge, die im Titel eines von mehreren angegebenen Wörtern (hier: die Wörter "in" und "of") enthalten, in die erste von zwei Zieldateien kopieren, die übrigen in die zweite Zieldatei.
- word_search_f.xml wie word_search.xml, mit Fehlern.
- words_search.xml Aus Datei inh die Beiträge, die im Titel alle angegebenen Wörter (hier: information und library) enthalten, in die erste von zwei Zieldateien kopieren, die übrigen in die zweite Zieldatei.

- words_search2.xml Aus Datei inh die Beiträge, in denen sowohl "information" oder "automation" als auch "library" vorkommt, in die erste von zwei Zieldateien kopieren, die übrigen in die zweite Zieldatei.
- title_length.xml Zahl der Wörter im Titel der Einträge in Datei inh feststellen, entsprechenden Kommentar + Titel ausgeben
- title_length_2.xml wie title_length.xml, andere Lösung
- satzz_pruf.xml Aus Datei jnh die Einträge, die nach dem/den Satzzeichen kein blank enthalten, in die erste, die übrigen in die zweite Zieldatei kopieren.
- atsf_pruf.xml Kopie der Datei atsf erstellen, in der die fehlerhaften Einträge markiert sind.
- atsf_pruf_2.xml Kopie der Datei atsf erstellen, in der die fehlerhaften Einträge durch entsprechende detaillierte Fehlerhinweise markiert sind (fehlende Kategorien; doppelte Kategorien; unbekannte Kategorien; falsche Reihenfolge).
- atsf_pruf_3.xml wie atsf_pruf_2.xml, Prüfung mit "text-contains-ordered" (ZF, ZFM, ZFH) statt mit "text-contains" (ZF+).
- atsf2xml.xml Die korrekten Einträge aus der Datei atsf.txt mit xml-Tags versehen und in die erste von zwei Zieldateien kopieren, die fehlerhaften Einträge mit detaillierten Fehlerhinweisen (fehlende Kategorien; doppelte Kategorien; unbekannte Kategorien; falsche Reihenfolge) versehen in die zweite Zieldatei kopieren.
- atsf2xml_b.xml Wie atsf2xml.xml; zusätzlich sind Leerzeilen und Zeilen, die mit blanks beginnen, Anfang einer Texteinheit.
- inh_von.xml Datei inh mit Hilfe von <text-part-start> und <text-part-end> umwandeln in: "Titel (von: Autoren)".
- inh_von2.xml Datei inh mit Hilfe von <brackets> umwandeln in: "Titel (von: Autoren)".
- ibam_silben2.xml In ibam.txt Wörter mit mehr als zwei Silben zählen, Anzahl am Zeilenende ausgeben.
- ibam_silben2_mrk.xml In ibam.txt Wörter mit mehr als zwei Silben zählen und markieren, Anzahl am Zeilenende ausgeben.
- ibam_silben.xml In ibamr.txt Zahl der Wörter, Silben und Elision pro Vers, Zahl der mehr als 2-silbigen Wörter im Vers; Durchschnittswerte erheben.

ibam_speech.xml	Aus ibamr.txt Passagen mit direkter Rede mit Gedicht- und Zeilennummer extrahieren (Lösung mit prepare-index und generate-index)
ibam_speech_t.xml	Aus ibamr.txt Passagen mit direkter Rede mit Gedicht- und Zeilennummer extrahieren (Lösung mit transform)
ibam_speech_tp.xml	Wie ibam_speech_t.xml, zusätzlich Anteil der direkten Rede (Zahl der Wörter) am gesamten Gedicht feststellen und ausgeben
bibl_aut1.xml	In der Bibliographie beim zweiten und evtl. weiteren Beiträgen vom gleichen Autor den Namen des Autors ersetzen durch "(id.)".
recker_dol.xml	Wörter mit tags <w>...</w> versehen (Aufgabe Recker in der TEI-Liste vom 26.4.2011, nach dem von Recker gewünschten Ergebnis.
recker_do.xml	wie recker_dol.xml, jedoch Satzzeichen nicht zum Wort zählen.
ged_xsl.xml	Demo-Skript zum Einbinden von xslt-Skripten in TXSTEP-Skripts: im TXSTEP-Skript Teil 1 Gedicht-Titel in Versalien umwandeln, dann mit xslt-Skript Datei in html-Datei umwandeln (dabei Autor- und Erscheinungsjahr-Angaben umbauen und umstellen), dann im TXSTEP-Skript Gedicht-Überschriften aus der html-Datei herauskopieren.
abs_lines.xml	Beim tag <abs> ein Attribut lines="n" einfügen, wobei n die Zahl der Zeilen im jeweiligen Abschnitt angibt (Beispiel für mode=from-end = "MODUS=R" in #KO).
ref-script.xml	"pattern matching across elements": Klammern in Quelldatei in tags <ref> bzw. </ref> verwandeln (aus <pre><p>This theory has been refuted (Jones <i>et al.</i> 2003).</p></pre> soll werden: <pre><p>This theory has been refuted <ref>Jones <i>et al.</i> 2003</ref>.</p></pre>
ref.xsl	dto., Lösung mit xslt.

2. Vergleichen und Weiterverarbeiten

2.1. TUSTEP-Dateien

ve.xml	Wortweises Vergleichen der Dateien schella.x und schellb.x, Erstellen eines Vergleichsprotokolls, Abspeichern der Unterschiede der beiden Dateien in der Syntax von Korrekturanweisungen des Tustep-Programms KAUSFUEHRE in der zu corrections angegebenen Datei; Wortlaut und Umgebung des Lemmas
--------	--

(= vom Unterschied betroffene Wörter der VersionA) und Stellenangabe der Varianten in VersionB mit ausgeben.

- ka.xml Erstellen einer Kopie (Name: schellk.x) der Datei schella.x, in der die Korrekturanweisungen der bei ve.xml zu corrections angegebenen Datei ausgeführt sind (die Kopie muss also inhaltlich identisch sein mit der bei ve.xml als versionB angegebenen datei schellb.x) und Nachprüfen des Ergebnisses durch Vergleich mit schellb.x.
- sep_diff.xml Die Unterschiede, die durch ve.xml in der Datei schell.diff abgespeichert wurden, zur Vorbereitung für einen kritischen Apparat sichten und aufteilen: Unterschiede, die nur Versalumlaute vs. Ae Oe Ue betreffen, in die Datei schellkv
Unterschiede, die nur Satzzeichen betreffen, in die Datei schellks
Unterschiede, die nur Orthograpica wie th und t, c und k, y ind i betreffen, in die Datei schellko
Andere Varianten in die Datei schellka (= Apparat).
- ve_tuscr.xml wie ve.xml, mit tustep- und tuscript-Einschüben. Soll nur die Möglichkeit des Zusammenspiel von TXSTEP, TUSTEP und TUSCRIPT zeigen.
- bau_cmp.xml Artikel 9 aus den "12 Artikel der Bauern" in 5 Fassungen: Vergleichen, zeilensynoptische Liste der Varianten erzeugen.
- bau_bau_cmp.xml wie bau_cmp.xml, jedoch 12 Fassungen.
- 2.2. XML-Dateien
- 2.2.1. Allgemein
- cmp.xml Wortweises Vergleichen der Dateien schella.xml und schellb.xml, Erstellen eines Vergleichsprotokolls, Abspeichern der Unterschiede in einer an TEI angelehnten Syntax, die alle Informationen enthält, die für die Korrekturanweisungen des TUSTEP-Programms KAUSFUEHRE notwendig sind.
- cmp_s.xml wie cmp.xml, aber von VersionB nur zwei Ausschnitte mit VersionA vergleichen.
- cmp_corr.xml Erstellen einer Kopie (Name: schellk.xml) der Datei schella.xml, in der die Korrekturanweisungen der bei cmp.xml zu corrections angegebenen Datei schell-diff.xml ausgeführt sind (die Kopie muss also inhaltlich identisch sein mit der bei cmp.xml als versionB angegebenen Datei schellb.xml).
- cmp_n.xml Wie cmp.xml, dabei großgeschriebene Umlaute als gleichwertig mit Ae, Oe, Ue behandeln (d.h. Wörter, die sich nur darin unterscheiden, nicht als Unterschiede ausgeben).

- vex.xml Wortweises Vergleichen der Dateien schella.xml und schellb.xml, dabei großgeschriebene Umlaute als gleichwertig mit Ae, Oe, Ue behandeln (d.h. Wörter, die sich nur darin unterscheiden, nicht als Unterschiede ausgeben), ebenso die Wörter "als" und "wie" als gleichwertig behandeln (nur als Demo, in der Praxis kaum sinnvoll); Erstellen eines Vergleichsprotokolls, Abspeichern der Unterschiede in der Syntax von Korrekturanweisungen des Tustep-Programms KAUSFUEHRE in der zu corrections angegebenen XML-Datei; Wortlaut und Umgebung des Lemmas (= vom Unterschied betroffene Wörter der VersionA) und Stellenangabe der Varianten in VersionB mit ausgeben. Ausgabe mit minimalem XML-markup. Eine Normalisierung von Texten nach diesem Muster zum Zweck des Vergleichs ist vor allem zur Kontrolle der Ergebnisse von automatischen Arbeitsschritten oder von Handeingriffen hilfreich. Für die halbautomatische Korrektur nach Doppel-Erfassung oder für das Erheben von Varianten als Vorbereitung für eine kritische Edition ist eine solche Normalisierung nur in Ausnahmefällen sinnvoll.
- vexx.xml wie vex.xml, jedoch Abspeichern der Unterschiede als Lesarten in einer an TEI angelehnten Syntax, die gleichzeitig alle Informationen enthält, die für die Korrekturanweisungen des Tustep-Programms KAUSFUEHRE notwendig sind.
- kax.xml Erstellen einer Kopie (schellk.xml) der Datei schella.xml, in der die Korrekturanweisungen der bei vex.xml oder vexx.xml zu corrections angegebenen Datei ausgeführt sind. Diese Kopie muss also inhaltlich identisch sein mit der beim Vergleichen als versionB angegebenen Datei schellb.xml, bis auf die Unterschiede, die bei vex.xml bzw. vexx.xml nicht erhoben wurden: Versal-Umlaute vs. Ae,Oe,Ue und "als" vs. "wie".
- vexx_2.xml Vergleichen der Datei schellb.xml mit der durch kax.xml erstellten Datei schellk.xml; in der zu corrections angegebenen Datei schellk-diff.xml dürfen jetzt nur Unterschiede bezüglich der Versal-Umlaute vs. vs. Ae,Oe,Ue und "als" vs. "wie" nachgewiesen sein.
- precorr.xml Beispiel für kontrolliertes Arbeiten:
Zur halbautomatischen Korrektur nach Doppelerfassung werden in der durch automatischen Vergleich erzeugten diff-Datei diejenigen Einträge (manuell) als Kommentar markiert, die in der ersten Fassung (VersionA) richtig sind.
Vor der (automatischen) Ausführung der Korrektur (z.B. durch kax.xml) werden die Einträge in zwei Dateien aufgeteilt: nicht-markierte Korrekturen und markierte. Nur erstere werden (hier nicht mehr gezeigt) für "correct" genutzt.

2.2.2 Zur Vorbereitung von kritischen Editionen

- cmp.xml (siehe oben); (weiter mit sep_variants.xml oder sep_variants_a2.xml).

- sep_variants.xml Die Unterschiede, die durch cmp.xml in der Datei schell-diff.xml abgespeichert wurden, als Vorbereitung für einen kritischen Apparat sichten und aufteilen: Unterschiede, die nur Versalumlaute vs. Ae Oe Ue betreffen, in die Datei schellkv.xml Unterschiede, die nur Satzzeichen betreffen, in die Datei schellks.xml Unterschiede, die nur Orthograpica wie th und t, c und k, y ind i betreffen, in die Datei schellko.xml Andere Varianten (Apparateinträge) in die Datei schellka.xml.
(Weiter mit insert_app.xml).
- sep_variants_2.xml dto., andere Lösung: wiederkehrende tables beim ersten Mal mit Namen versehen, bei späterer Verwendung über Namen referenzieren. (weiter mit insert_app.xml).
- cmp-list.xml Liste der mit sep_variants für den Apparat ausgewählten Lesarten, Ausgabe jeweils zeilensynoptisch unter dem Text der Version A.
- insert_app.xml Die Einträge aus schellka.xml zu Apparateinträgen umbauen und über CORRECT in den Text der VersionA (Ergebnis: schella_app.xml) einsetzen; diese Datei mit der externen TUSTEP-Prozedur schellsatz setzen und als PostScript-Datei ausgeben.

Die Datei schellka.xml bedarf noch der philologischen Überarbeitung. Eine solche ist für insgesamt drei Stellen beispielhaft in der Datei /uebq*schellkac.xml vorgenommen: für Umstellungen sind die von cmp.xml gefunden und mit sep_variants.xml ausgewählten wortgenauen Varianten zu philologisch sinnvollerer Einheiten zusammengefasst. (In insert_app.xml ist diese Datei bereits als Kommentar zur Definition der Variablen "variants" enthalten).
- schellsatz Vorbereitete Prozedur zum Satz der Edition, wird in insert_app.xml etc. benutzt.
- check_var.xml Zum Prüfen der manuell durchgeführten Änderungen mit COLLATE eine zeilensynoptische Liste VersionA-VersionB ausgeben.
- sep_variants_a2.xml Wie sep_variants.xml, jedoch die ersten drei o.g. Typen von Varianten mit dem Attribut typ="umlaut"|"punctuation"|"orthographic" beim tag <rdg> versehen, alle außer typ="umlaut" in die Datei schellka.xml schreiben (weiter mit insert_app_a2.xml oder insert_app_a22.xml).
- insert_app_a2.xml Wie insert_app.xml, jedoch drei Apparate, ausgehend vom Ergebnis von sep_variants_a2.xml: außer dem Hauptapparat je einen eigenen Apparat für Varianten vom Typ "orthographic" und vom Typ "punctuation".

insert_app_a22.xml

dto., andere Lösung in pass "check-lemm" und
"more-word-lemma": reassemble statt modify.

insert_app_a2_numerg.xml

Wie insert_app_a2.xml, jedoch wird davon ausgegangen, dass
die Korrekturdatei verkürzte Stellenangaben enthält (=
Voreinstellung von #VE, nicht jedoch von <compare>).

- mehr als 2 Textzeugen:

cmp3.xml

Vergleichen von mehr als 2 Textzeugen (hier: schella.xml
mit schellb.xml und mit schellc.xml), anschließend
kumuliertes Vergleichsprotokoll (weiter mit
sep_variants_3.xml).

sep_variants_3.xml

wie sep_variants_a2, jedoch für mehr als zwei Zeugen;
zusätzliche Schritte notwendig zum Kumulieren und
Sortieren der Varianten (weiter mit insert3_app.xml).

insert3_app.xml

Wie insert_app_a2.xml, für mehr als 2 Textzeugen.
Zusätzliche Schritte notwendig zum Zusammenfassen der
Varianten zur gleiche Lemma-Stelle u.a. (siehe die
Kommentare im Skript).

cmp3_b.xml

wie cmp3.xml, aber context-for-parcelling="1" (gibt beim
Vergleich VersionA-VersionC an einigen Stellen bessere
Zuordnung Lemma-Reading; weiter mit sep_variants_3b.xml).

sep_variants_3b.xml

wie sep_variants_3.xml, aber Varianten aus cmp3_b.xml
(weiter mit insert3_app.xml).

cmp3pars.xml

wie cmp3.xml (ohne Protokoll), aber mit einer mehrfach
verwendbaren "procedure".

cmp3def.xml

wie cmp3.xml (ohne Protokoll), aber mit einer mehrfach
verwendbaren "procedure" und definierten Dateinamen.

- Ausgabe in html

schell-html_cmp.xml

Version a mit b und c sowie b mit a und c mit a
vergleichen, damit auch von b bzw. c aus auf die Version
a verwiesen werden kann. Aufteilen der Unterschiede wie
in sep_variants (siehe oben), Sortieren der relevanten
Varianten nach Stelle, Art, Wortlaut, Zeuge.

schell-html_build.xml

html-Fassung der Edition erzeugen:
Schirm vertikal aufgeteilt in zwei Hälften, linke Hälfte
= Version A mit Hervorhebung der Stellen, an der Stellen,
an der Varianten vorliegen, und mit Links zum darunter
stehenden Apparat. Von dort über Zeugen-Code Links in die
rechte Hälfte, in der die betr. Version angezeigt wird
mit Hervorhebung der Stellen, an denen sie von Version A
abweicht, und Links zum darunter stehenden Apparat. (Die

frames für die Bildschirm-Aufteilung und die Überschrift-Zeilen für Text und Apparat stehen in /ueb/schell bereit.)

3. Register-Arbeit, Sortieren:

autorenliste.xml	alphabetische Liste der Autoren aus der Datei inh.xml als xml-Datei, Seitenzahlen als Referenz.
autorenliste_rel.xml	wie autorenliste.xml, mit relativen Pfadangaben
autreg_inh.xml	alphabetische Liste der Autoren + Seitenzahlen aus der Tustep-Datei inh als Tustep-Datei, xml-tags <eintrag> ... </eintrag> um jede Zeile. Zieldatei nicht überschreiben, sondern neue Liste hinten anhängen.
inh_alpha.xml	Einträge der Datei inh.xml alphabetisch nach dem ersten Autor sortieren; bei mehreren Autoren: Verweiseinträge vom zweiten und dritten Autor auf den ersten erzeugen und einsortieren.
inh_alpha2.xml	wie inh_alpha.xml, andere Lösung (brackets statt text-part-start...).
inhx_stichw.xml	Alphabetisches Verzeichnis aller mit t beginnenden Titelstichwörter aus der Datei inh.xml.
rv_greek_s.xml	alphabetischer Wortformenindex zur Datei lucas.
rv_greek_x.xml	dto., aber zu Datei lucas.xml; Im Register: Einträge, die x enthalten, in Versalien ausgeben.
rv_greek_xs.xml	dto., ohne Umwandlung in Versalien, mit Satzausgabe über eingebundenes TUSTEP-Skript.
rv_greek_xr.xml	rückläufiger Wortformenindex zu Datei lucas.xml.
rv_greek_xr1.xml	dto., Ausgabe als Liste.
ibamr_metren.xml	Verzeichnis der metrischen Schemata (Folge von Daktylen und Spondeen; Zeichenfolge hinter @) in Datei ibam bzw. ibamr.
ibamr_metren_rf.xml	wie ibamr_metren.xml, zusätzlich relative Häufigkeit ausgeben, außerdem Liste der Schemata nach absteigender Häufigkeit ausgeben.
ibamr_metren2.xml	Verzeichnis der metrischen Schemata (Folge von Daktylen und Spondeen; Zeichenfolge hinter @ in Datei ibam bzw. ibamr), sortiert nach Zahl der Daktylen (= "langsame" Verse nach vorn).
ibamr_freq.xml	zwei Verzeichnisse der in Datei ibamr vorkommenden Wörter: 1. alphabetisch sortiert, mit Referenzen 2. nach Häufigkeiten sortiert, ohne Referenzen.
ibams_freq.xml	dto., aber Quelle = ibamr.txt (Fremd-Datei) mit ausführlichen Kommentaren im Skript.

ibamrs_freq.xml	wie ibamr_freq.xml, aber alphabetisches Verzeichnis aufgeteilt auf zwei Dateien: hapax legomena und mehrfach vorkommende Wörter.
ibamr_kwic.xml	KWIC-Index zu dem Gedicht in ibamr; Kontext: jeweils eine Gedichtzeile.
ins.xml	Personenregister (mit Referenzen) anreichern durch Kurzinfos zu den Personen.
ed_bibl3.xml	Die beiden Teile der Bibliographie bibl.txt jeweils nach Erscheinungsdatum sortieren.
ed_bibl3_ext.xml	wie ed_bibl3.xml, jedoch die dort erstellten external procedures benutzen.
ed_bibl3see.xml	Die beiden Teile der Bibliographie bibl-see.txt jeweils nach Erscheinungsdatum sortieren, neu durchnummerieren und Verweise aktualisieren.
kol_aut.xml	Verzeichnis der Referenten der Tübinger EDV-Kolloquien erstellen; Referenz: Kolloquiums-Nummer und Datum.
import_rtf.xml	Register zu kurzer (aus .docx erzeugter) rtf-Datei, Referenzen = aus dem Text (zwei durch "." getrennte Zahlen, in ((....)) am Zeilenanfang).
import_rtf-rec.xml	wie import_rtf.xml, aber laufende Nummer der Zeile als Referenz.
import_rtf-std.xml	wie import_rtf-rec.xml, explizite Angabe des Sortierschlüssels in <sort>.

4. Prozeduren definieren und ausführen

procdef.xml	Muster für Definition und Ausführen von mehrfach verwendbaren procedures.
procext.xml	Muster für Benutzung der in procdef.xml definierten (jetzt externen) procedure.
cmp3pars.xml	wie cmp3.xml (ohne Protokoll), mit einer mehrfach verwendbaren procedure (siehe oben unter 2.2.2)
cmp3def.xml	wie cmp3.xml (ohne Protokoll), mit einer mehrfach verwendbaren procedure und definierten Dateinamen (siehe oben unter 2.2.2).

5. Leerer Rahmen für TXSTEP-Skript

txstep_frame.xml	Datei, die als Rahmen für ein TXSTEP-Skript dienen kann.
------------------	--

Weitere Infos: <http://www.tustep.uni-tuebingen.de/txstep.html>
<http://www.txstep.de>