

Für die TXSTEP-Version März 2021 stehen folgende Dateien und Test-Skripts 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>...</a> <t>...</t> <s>...</s> </beitrag> ... </inhalt>
inhb5.xml	wie inh.xml, jedoch bei mehrfach-Autoren jeder Autor zwischen <a> und </a>.
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 Editionswesen &.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
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  
 schell.diffi        Beispiel einer Quelldatei für precorr.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-Skripts (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.).  
                       Ziel-Datei: inhb5.xml  
 inh\_f2xml.xml      dto. für inh.txt (= inh im "Fremddaten-Format").  
                       Ziel-Datei: inhb5.xml  
 inh2xml\_2.xml      wie inh2xml.xml, andere Lösung.  
                       Ziel-Datei: inhb5.xml  
 inh\_f2xml\_2.xml    dto. für inh.txt (= inh im "Fremddaten-Format").  
                       Ziel-Datei: inhb5.xml

- inh2xml\_2n.xml wie inh2xml\_2.xml, aber Autoren nummerieren.  
Ziel-Datei: inhb5.xml
- inh2xml\_a.xml wie inh2xml.xml, dabei gleichzeitig  
Autor, Adam in Adam AUTOR umwandeln.  
Ziel-Datei: inhb5.xml
- inh\_lnr.xml Datei inh.xml: vor den Autornamen dreistellige  
laufende Nummer einfügen.  
Ziel-Datei: inh-lnr
- xml2inh.xml Datei inh.xml: Tags entfernen, je Eintrag eine Zeile  
nach dem Muster der Datei inh (s.o.).  
Ziel-Datei: inh.tstp
- 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>...  
Ziel-Datei: inh\_length.txt
- inh\_length\_2.xml wie inh\_length.xml, aber (<insert-at-end>...  
Ziel-Datei: inh\_length.txt
- inh\_length\_3.xml wie inh\_length\_2.xml, aber <brackets><opening-bracket>  
statt <read-values-from>.  
Ziel-Datei: inh\_length.txt
- inh\_length\_3b.xml wie inh\_length\_3.xml, aber zusätzlich <closing-bracket>.  
Ziel-Datei: inh\_length.txt
- inh\_length\_4.xml wie inh\_length.xml, aber nur wenn nach : "information"  
oder nach ; "Autor"; im Titel "e" vor "information"  
zu "E" austauschen.  
Ziel-Datei: inh\_length.txt
- inh\_length\_mrk.xml wie inh\_length.xml, zusätzlich den kürzesten  
und den längsten entsprechend markieren.  
Ziel-Datei: inh\_length.txt
- inh\_length\_proz.xml Datei inh.txt: Länge der einzelnen Beiträge  
und Anteil (in Prozent) an der Gesamtlänge errechnen  
und einsetzen, Gesamt-Seitenzahl ausgeben  
Ziel-Datei: inh\_length.txt
- 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>".  
Ziel-Datei: inh\_tit.xml
- 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>.  
Ziel-Datei: inh\_nampag.xml
- names\_pages2.xml wie names\_pages.xml, andere Lösung (AKn/EKn statt nur XX).  
Ziel-Datei: inh\_nampag.xml

- names\_pages3.xml wie names\_pages2.xml, andere Lösung  
( (Kn/)Kn statt AKn/EKn ).  
Ziel-Datei: inh\_nampag.xml
- names\_pages3b.xml wie names\_pages3.xml, (Kn/)Kn mit eliminate statt select.  
Ziel-Datei: inh\_nampag.xml
- 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>  
Ziel-Datei: inh\_nampag.xml
- 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>,<br><givenName>Bart E.</givenName></name><br><pages>5-26</pages><br></item>  
Ziel-Datei: inh\_nampag.xml
- name\_pages\_2.xml wie name\_pages.xml, andere Lösung (ETE statt XX  
in Schleife für jeden Autor).  
Ziel-Datei: inh\_nampag2.xml
- vor\_famnam.xml Kopie der Datei inh erstellen, in der die  
Vornamen der Autoren vor den in Versalien  
umgewandelten Familiennamen stehen.  
Ziel-Datei: inh\_sel.tstp
- aut\_ref.xml Verweise vom zweiten und weiteren Autoren (TA+, 2 Dg.)  
auf den ersten Autor erzeugen, z.B.  
Verfasser, Zeno see Autor, Adam  
Ziel-Datei: inhalt\_ref.txt
- aut\_ref\_2.xml wie aut\_ref.xml, andere (kürzere) Lösung (KEN, KSP).  
Ziel-Datei: inhalt\_ref.txt
- aut\_ref\_3.xml wie aut\_ref.xml, dritte Lösung  
(ZFZ/ZSP statt ZF+ in Schleife).  
Ziel-Datei: inhalt\_ref.txt
- snr\_rest.xml Kopie der Datei inh erstellen, in der die  
Seitenangaben vor den Autorennamen stehen.  
Ziel-Datei: inh\_sel.tstp
- snr\_rest\_2.xml wie snr\_rest.xml, andere Lösung  
(subunits statt text-parts);  
Titel nicht mit ausgeben.  
Ziel-Datei: inh\_sel.tstp
- 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.  
Ziel-Dateien: inh\_sel.tstp und inh\_notsel.tstp

- 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.  
Ziel-Dateien: inh\_sel.tstp und inh\_notssel.tstp
- 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.  
Ziel-Dateien: inh\_sel.tstp und inh\_notssel.tstp
- title\_length.xml Zahl der Wörter im Titel der Einträge in Datei inh feststellen, entsprechenden Kommentar + Titel ausgeben  
Ziel-Datei: inh\_sel.tstp
- title\_length\_2.xml wie title\_length.xml, andere Lösung  
Ziel-Datei: inh\_sel.tstp
- 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.  
Ziel-Dateien: inh\_ok.tstp und inh\_false.tstp
- atsf\_pruf.xml Kopie der Datei atsf erstellen, in der die fehlerhaften Einträge markiert sind.  
Ziel-Datei: atsf.tstp
- 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).  
Ziel-Datei: atsf.tstp
- atsf\_pruf\_3.xml wie atsf\_pruf\_2.xml, Prüfung mit "text-contains-ordered" (ZF, ZFM, ZFH) statt mit "text-contains" (ZF+).  
Ziel-Datei: atsf.tstp
- 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.  
Ziel-Dateien: atsf.xml und atsf\_err.txt
- atsf2xml\_b.xml wie atsf2xml.xml; zusätzlich sind Leerzeilen und Zeilen, die mit blanks beginnen, Anfang einer Texteinheit.  
Ziel-Dateien: atsf.xml und atsf\_err.txt

- inh\_von.xml Datei inh mit Hilfe von <text-part-start> und <text-part-end> umwandeln in:  
"Titel (von: Autoren)".  
Ziel-Datei: inhx
- inh\_von2.xml Datei inh mit Hilfe von <brackets> umwandeln in:  
"Titel (von: Autoren)".  
Ziel-Datei: inhx
- ibam\_silben2.xml In ibam.txt Wörter mit mehr als zwei Silben zählen, Anzahl am Zeilenende ausgeben.  
Ziel-Datei: ibamt\_w
- ibam\_silben2\_mrk.xml In ibam.txt Wörter mit mehr als zwei Silben zählen und markieren, Anzahl am Zeilenende ausgeben.  
Ziel-Datei: ibamt\_wm
- 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.  
Ziel-Datei: ibamt\_w
- ibam\_speech.xml Aus ibamr.txt Passagen mit direkter Rede mit Gedicht- und Zeilennummer extrahieren (Lösung mit prepare-index und generate-index)  
Ziel-Datei: ibam\_speeches.xml
- ibam\_speech\_t.xml Aus ibamr.txt Passagen mit direkter Rede mit Gedicht- und Zeilennummer extrahieren (Lösung mit transform)  
Ziel-Datei: ibam\_speeches\_to.xml
- ibam\_speech\_tp.xml wie ibam\_speech\_t.xml, zusätzlich Anteil der direkten Rede (Zahl der Wörter) am gesamten Gedicht feststellen und ausgeben  
Ziel-Datei: ibam\_speeches\_t.xml
- bibl\_aut1.xml In der Bibliographie beim zweiten und evtl. weiteren Beiträgen vom gleichen Autor den Namen des Autors ersetzen durch "(id.)".  
Ziel-Datei: bibl-aut1.txt
- 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.)  
Ziel-Datei: recker\_wrap1.xml
- recker\_do.xml wie recker\_dol.xml, jedoch Satzzeichen nicht zum Wort zählen.  
Ziel-Datei: recker\_wrap.xml
- 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).  
Ziel-Datei: schell-abslines.xml

2. Vergleichen und Weiterverarbeiten2.1. 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.  
Ziel-Dateien: schell-diff, schell-diff.list
- 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 abgegebenen datei schellb.x) und Nachprüfen des Ergebnisses durch Vergleich mit schellb.x.  
Ziel-Dateien: schellk, schell-corr.list, schell-comp.list
- precorr.xml              Beispiel für kontrolliertes Arbeiten:  
Zur halbautomatischen Korrektur nach Doppelerfassung werden in einer Kopie der durch automatischen Vergleich erzeugten diff-Datei (hier in schell-diffi = einer Kopie der mit dem Skript ve.xml erstellten Tustep-Datei schell-diff) die Einträge, die in der ersten Fassung (version-a) korrekt transkribiert sind, mit einem "-" am Zeilenanfang markiert. Einträge, in denen beide Fassungen falsch transkribiert wurden, werden mit "-+" am Zeilenanfang markiert und durch eine danach eingefügte Zeile mit der korrekten Korrekturanweisung ersetzt.  
  
Danach werden mit diesem Skript vor der Ausführung der Korrektur (z.B. durch ein Skript wie ka.xml) die die nicht-markierten und die markierten Einträge auf zwei Dateien aufgeteilt. Nur die Datei mit den nicht-markierten bzw. mit den ergänzten Einträgen wird für die automatische Korrektur genutzt. Die Datei mit den markierten Einträgen kann bei Bedarf zur Dokumentation bzw. Kontrolle der durchgeführten Arbeitsschritte dienen.  
Ziel-Dateien: schell-diffk.xml, schell-diffx.xml  
  
(Bei Korrektur-Dateien im xml-Format, die mit Skripts wie vex.xml oder vexx.xml erzeugt wurden, ist die Aufteilung auf zwei Dateien überflüssig: Zeilen, die nicht auszuführende Korrekturen enthalten, können dort mit <!-- ... --> als xml-Kommentar gekennzeichnet werden; diese werden von <correct> übergangen.)
- var\_lev.xml              Zu den mit ve.xml erhobenen und in schell-diff gespeicherten Unterschieden die Levenshtein-Distanz ergänzen und in der Reihenfolge ihres Vorkommens sowie sortiert nach aufsteigendem Distanzwert



- ausgeben  
Ziel-Dateien: diff-lev, diff-lev\_s
- 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 und i betreffen, in die Datei schellko  
Andere Varianten in die Datei schellka (= Apparat).  
Ziel-Dateien: schellka, schellkv, schellks, schellko
- ve\_tuscr.xml wie ve.xml, mit tustep- und tuscript-Einschüben.  
Soll nur die Möglichkeit des Zusammenspiels von TXSTEP, TUSTEP und TUSCRIPT zeigen.  
Ziel-Dateien: schell-diff, schell-diff.list
- bau\_cmp.xml Artikel 9 aus den "12 Artikel der Bauern" in 5 Fassungen: Vergleichen, zeilensynoptische Liste der Varianten erzeugen.  
Ziel-Dateien: bauu-diff2, bau-diff2s, schell-diff.list
- bau\_bau\_cmp.xml wie bau\_cmp.xml, jedoch 12 Fassungen.  
Ziel-Dateien: bau-diff12, bau-diff12.list
- 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.  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- cmp\_s.xml wie cmp.xml, aber von VersionB nur zwei Ausschnitte mit VersionA vergleichen.  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- 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).  
Ziel-Dateien: schellk.xml, schell-diffk.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).  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- 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.

Ziel-Dateien: schell-diff.xml, schell-diff.list

- 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.  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- 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".  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- 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.  
Ziel-Dateien: schellk-diff.xml, schell-diff.list

### 2.2.2 Zur Vorbereitung von kritischen Editionen

- cmp.xml (siehe oben); (weiter mit sep\_variants.xml oder sep\_variants\_a2.xml).  
Ziel-Dateien: schell-diff.xml, schell-diff.list
- var\_x\_lev.xml Zu den mit cmp.xml erhobenen und in schell-diff.xml gespeicherten Unterschieden die Levenshtein-Distanz ergänzen und in der Reihenfolge ihres Vorkommens ausgeben.  
Ziel-Datei: schell\_diffab-lev.xml
- var\_xx\_lev.xml wie var\_x\_lev.xml, jedoch wird die Quelldatei nicht element-weise, sondern record-weise gelesen.  
Die Levenshtein-Distanz wird am Zeilenende als empty

- element mit dem Attribut `dist="n"` angehängt.  
Ziel-Datei: `schell_diff2-lev.xml`
- `var_xxs_lev.xml` wie `var_xx_lev.xml`, jedoch werden die Einträge in der Ziel-Datei nach aufsteigender Levenshtein-Distanz sortiert.  
Ziel-Datei: `schell_diffe-lev_s.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` und `i` betreffen, in die Datei `schellko.xml`  
Andere Varianten (Apparateinträge) in die Datei `schellka.xml`.  
(Weiter mit `insert_app.xml`).  
Ziel-Dateien: `schellka.xml`, `schellkv.xml`, `schellks.xml`, `schellko.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`).  
Ziel-Dateien: `schellka.xml`, `schellkv.xml`, `schellks.xml`, `schellko.xml`
- `cmp-list.xml` Liste der mit `sep_variants` für den Apparat ausgewählten Lesarten, Ausgabe jeweils zeilensynoptisch unter dem Text der Version A.  
Ziel-Datei: `schell-redlist`
- `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.  
Ziel-Dateien: `schella_app.xml`, `schellinst.xml`, `schell_typo.xml`; `schella.ps` (Satzausgabe)
- Die Datei `schellka.xml` bedarf noch der editorischen 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 sinnvolleren 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.  
Ziel-Dateien: `schellkac.xml`, `schella_red.lst`

- 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`).  
Ziel-Dateien: `schellka.xml`, `schellkv.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"`.  
Ziel-Dateien: `schella_app.xml`, `schellinst.xml`,  
`schell_typo.xml`; `schella.ps` (Satzausgabe)
- insert\_app\_a22.xml dto., andere Lösung in pass `"check-lemm"` und `"more-word-lemma"`: `reassemble` statt `modify`.  
Ziel-Dateien: `schella_app.xml`, `schellinst.xml`,  
`schell_typo.xml`; `schella.ps` (Satzausgabe)
- 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>`).  
Ziel-Dateien: `schella_app.xml`, `schellinst.xml`,  
`schell_typo.xml`; `schella.ps` (Satzausgabe)
- 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`).  
Ziel-Dateien: `schell-diffab.xml`, `schell-diffac.xml`,  
`schell_diff.list`
- 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`).  
Ziel-Dateien: `schellka.xml`, `schellkv.xml`, `schellkas.xml`,  
`schell-pars`
- 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).  
Ziel-Dateien: `schella_app.xml`, `schellinst.xml`,  
`schell_typo.xml`; `schella.ps` (Satzausgabe)
- 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`).  
Ziel-Dateien: `schell-diffab.xml`, `schell-diffac.xml`,  
`schell_diff.list`
- sep\_variants\_3b.xml wie `sep_variants_3.xml`, aber Varianten aus `cmp3_b.xml`  
Ziel-Dateien: `schellka.xml`, `schellkv.xml`, `schellkas.xml`,  
`schell-pars`  
(weiter mit `insert3_app.xml`).

- cmp3pars.xml wie cmp3.xml (ohne Protokoll), aber mit einer mehrfach verwendbaren "procedure".  
Ziel-Dateien: schell-diffab.xml, schell-diffac.xml, schell\_diff.list
- cmp3def.xml wie cmp3.xml (ohne Protokoll), aber mit einer mehrfach verwendbaren "procedure" und definierten Dateinamen. Zusätzlich auch Vergleich von schellb.xml und schellc.xml mit schella.xml.  
Ziel-Dateien: schell-diffab.xml, schell-diffac.xml, schell\_diffba.xml, schell\_diffca.xml
- 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.)
- Die Datei /ueb/schell/fra3.html sollte mit Microsoft Edge oder Google Chrome angezeigt werden; Firefox und der Internet Explorer unterstützen die hier benutzten Elemente <frameset> und <frame> nicht mehr.

### 3. Register-Arbeit, Sortieren:

- autorenliste.xml alphabetische Liste der Autoren aus der Datei inh.xml als xml-Datei, Seitenzahlen als Referenz.  
Ziel-Datei: autlist.xml
- autorenliste\_rel.xml wie autorenliste.xml, mit relativen Pfadangaben  
Ziel-Datei: autlist.xml
- 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.  
Ziel-Datei: aut-inh
- 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.  
Ziel-Datei: inh\_alpha.xml

inh_alpha2.xml	wie inh_alpha.xml, andere Lösung (brackets statt text-part-start...) Ziel-Datei: inh_alpha.xml
inhx_stichw.xml	Alphabetisches Verzeichnis aller mit t beginnenden Titelstichwörter aus der Datei inh.xml. Ziel-Datei: stichwreg.xml
rv_greek_s.xml	alphabetischer Wortformenindex zur Datei lucas. Ziel-Datei: lucas-reg.xml
rv_greek_x.xml	dto., aber zu Datei lucas.xml; Im Register: Einträge, die x enthalten, in Versalien ausgegeben. Ziel-Datei: lucas-reg.xml
rv_greek_xs.xml	dto., ohne Umwandlung in Versalien, mit Satzausgabe über eingebundenes TUSTEP-Skript. Ziel-Dateien: lucas-reg.xml, reg.ps
rv_greek_xr.xml	rückläufiger Wortformenindex zu Datei lucas.xml. Ziel-Datei: lucas-reg.xml
rv_greek_xrl.xml	dto., Ausgabe als Liste. Ziel-Datei: lucas.list
ibamr_metren.xml	Verzeichnis der metrischen Schemata (Folge von Daktylen und Spondeen; Zeichenfolge hinter @) in Datei ibam bzw. ibamr. Ziel-Datei: ibammet.xml
ibamr_metren_rf.xml	wie ibamr_metren.xml, zusätzlich relative Häufigkeit ausgeben, außerdem Liste der Schemata nach absteigender Häufigkeit ausgeben. Ziel-Dateien: ibammet.xml und ibammetf.xml
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). Ziel-Datei: ibammet.xml
ibamr_freq.xml	zwei Verzeichnisse der in Datei ibamr vorkommenden Wörter: 1. alphabetisch sortiert, mit Referenzen 2. nach Häufigkeiten sortiert, ohne Referenzen. Ziel-Dateien: ibamr-wl.xml und ibamr-wlh.xml
ibams_freq.xml	dto., aber Quelle = ibamr.txt (Fremd-Datei) mit ausführlichen Kommentaren im Skript. Ziel-Dateien: ibamr-wl.xml und ibamr-wlh.xml
ibamrs_freq.xml	wie ibamr_freq.xml, aber alphabetisches Verzeichnis aufgeteilt auf zwei Dateien: hapax legomena und mehrfach vorkommende Wörter. Ziel-Dateien: ibamr-wl1.xml, ibamr-wl2.xml, ibamr_wlh.xml
ibamr_kwic.xml	KWIC-Index zu dem Gedicht in ibamr; Kontext: jeweils eine Gedichtzeile. Ziel-Dateien: ibamr-wl.xml und ibamr-kwic (für's Listing)

ins.xml	Personenregister (mit Referenzen) anreichern durch Kurzinfos zu den Personen. Ziel-Dateien: index_persons.xml, biographies.xml:
ed_bibl3.xml	Die beiden Teile der Bibliographie bibl.txt jeweils nach Erscheinungsdatum sortieren. Ziel-Datei: bibl_year.xml
ed_bibl3_ext.xml	wie ed_bibl3.xml, jedoch die dort erstellten external procedures benutzen. Ziel-Datei: bibl_year.xml (mit display der Zwischen-Datei zws1)
ed_bibl3see.xml	Die beiden Teile der Bibliographie bibl-see.txt jeweils nach Erscheinungsdatum sortieren, neu durchnummerieren und Verweise aktualisieren. Ziel-Datei: bibl_see_year.html (mit display von bibl-see_year_orignum.txt und Zwischen-Dateien)
kol_aut.xml	Verzeichnis der Referenten der Tübinger EDV-Kolloquien erstellen; Referenz: Kolloquiums-Nummer und Datum. Ziel-Datei: kol_autreg.xml
import_rtf.xml	Register zu kurzer (aus .docx erzeugter) rtf-Datei, Referenzen = aus dem Text (zwei durch "." getrennte Zahlen, in ((...)) am Zeilenanfang). Ziel-Datei: word_index.tsf
import_rtf-rec.xml	wie import_rtf.xml, aber laufende Nummer der Zeile als Referenz. Ziel-Datei: word_index
import_rtf-std.xml	wie import_rtf-rec.xml, explizite Angabe des Sortierschlüssels in <sort>. Ziel-Datei: word-index.tsf

#### 4. Prozeduren definieren und ausführen

procdef.xml	Muster für Definition und Ausführen von mehrfach verwendbaren procedures. Ziel-Dateien: sel.tu für Prozedur ibamsel für das Ergebnis
procext.xml	Muster für Benutzung der in procdef.xml definierten (jetzt externen) procedure. Ziel-Dateien: ibamsel.txt, inhsel.txt (Aufruf der Prozedur mit verschiedenen Dateien)
cmp3pars.xml	wie cmp3.xml (ohne Protokoll), mit einer mehrfach verwendbaren procedure (siehe oben unter 2.2.2) Ziel-Dateien: compare.tu für Prozedur schell-diffab.xml, schell-diffac.xml, Listing: schell-diff.list
cmp3def.xml	wie cmp3.xml (ohne Protokoll), mit einer mehrfach verwendbaren procedure und definierten Dateinamen (siehe oben unter 2.2.2). Ziel-Dateien: compare.tu für Prozedur

schell-diffab.xml, schell-diffac.xml,  
schell-diffba.xml, schell-diffca.xml

#### 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>