Anleitung und technische Details
Version 2.0

Von Ismail Prada

27. September 2017

Inhaltsverzeichnis

1 Datenformate

1.1 Standard-XML: RoXML

Die Daten des romanischen Seminars werden standardmässig in einem XML-Format gespeichert. Eine solche Standardisierung erleichtert die Umwandlung in andere Datenformate und das Suchen im Korpus.
XML-Dokumente sind in Baumstrukturen angelegt. Das bedeutet, dass von einer sogenannten Wurzel Zweige ausgehen, die sich jeweils in weitere Zweige teilen (Kinder genannt).
Die oberste Ebene in unserem Standardformat ist der < *corpus* >. Diesem untergeordnet sind die < *document* >-Elemente. Diese beziehen sich auf die als Input gelesenen Dokument und tragen diese Informationen als Attribute. Den Dokumenten wiederum untergeordnet sind die < *phrase* >-Elemente. Diese beinhalten einerseits den originalen, unbearbeiteten Satz, einen Verweis auf die Metadaten, sowie Informationen zu Eigenschaften des Satzes (zum Beispiel Vollständigkeit). Im Falle einer tokenisierten Datei (getaggt oder ungetaggt) finden sich als Kinder des < *phrase* >-Elements < *token* >-Elemente. Diese Elemente stellen zusammen nicht nur den tokenisierten Satz dar, sie enthalten auch Informationen zu jedem einzelnen Token. Üblicherweise sind dies Lemmata, POS und Dependenzen innerhalb des Satzes, je nach Vorhaben können hier aber auch mehr, weniger oder andere Informationen gespeichert sein.
Die meisten Applikationen benutzen nur die < *phrase* > und < *token* >-Ebenen.

1.1.1 Die Metadaten

Getrennt von den Sätzen werden die Metadaten festgehalten. Auch die Metadaten werden als XML-Dokument, also in Baumformat abgespeichert. Die oberste Ebene bildet wieder die Wurzel, deren Namen je nach Herkunft variieren kann (im Falle der CNDH-Daten ist sie zum Beispiel *CNDH* genannt). Der Wurzel untergeordnet sind $<$ *publication* $>$-Elemente, die ihre Id als Information enthalten. Eine Ebene weiter wiederum sind die anderen Informationen zu der Publikation zu finden:
$<$ *name* $>$ (Name der Publikation), $<$ *author* $>$ (Autor und Autoren-Id), $<$ *nation* $>$ (Land der Publikation), $<$ *publisher* $>$ (Verleger, Verlagsort und Verlagsjahr) und $<$ *year* $>$ (Das originale Erscheinungsjahr). Da die verfügbaren Metadaten je nach Korpus aber stark variieren, sind häufig auch andere Informationen zu finden.

1.2 WebAnno-TSV

Das Webanno-TSV-Format (TSV = *Tab Separated Values*) wird nicht als permanentes Format genutzt, stellt aber das übliche Import- und Exportformat von Webanno dar.
Die Kopfzeile definiert den Inhalt der Spalten. Hierbei verhält sich das Format jedoch speziell, da es sämtliche Spaltendefinitionen (auch *fieldnames* genannt) in die erste Spalte der ersten Zeile schreibt, sie aber durch Hashtags trennt.
Darunter folgen die einzelnen Sätze. Die Sätze werden jeweils mit einem Kommentar eingeleitet, der die Webanno-Id des Satzes festlegt. In der Zeile darunter folgt üblicherweise (aber nicht zwingend) ein Kommentar, eingeleitet mit *text=*, der den Satz als Ganzes darstellt. Erst eine Zeile danach beginnen die Informationen zu den einzelnen Token. Jede Zeile steht für ein Token und jede Spalte steht üblicherweise für ein Attribut im Standardformat.

1.2.1 Metadata

Etwas speziell verhält es sich mit der Spalte *Metadata*. Diese Spalte existiert nur zu Umwandlungszwecken. Als Information enthält sie das Dokument, aus dem der Satz stammt, sowie die Satz-Id des Satzes im Standardformat. Diese Informationen ermöglichen nach einem Bearbeitungsprozess in Webanno die Rückführung der Sätze in die Datei im Standardformat, ohne dass eine neue Datei geschrieben werden muss die exklusiv die bearbeiteten Sätze enthält.

2 Arbeitsablauf

2.1 Allgemein

Die gesamte Bearbeitung der Dateien sollte von denselben Personen durchgeführten werden können, welche die Daten auch wissenschaftlich evaluieren (ohne Hilfe eines technischen Mitarbeiters). Deshalb ist darauf geachtet worden, alle Programme als graphische Interfaces zu implementieren, die möglichst einfach zu bedienen sein sollten.
Der allgemeine Arbeitsablauf gestaltet sich folgendermassen:
Die Rohdaten werden zum Standardformat umgewandelt und als solches gespeichert. Aus dieser Form können sie entweder ungetaggt ins Webanno eingespeist werden, oder durch einen Tagger gespeist werden, der die Daten tokenisiert, taggt und sogar Dependenzen zuordnet. Solche getaggte Daten werden ebenfalls im Standardformat gespeichert (das Format wird dazu nur durch die Ebene $<$ *token* $>$ erweitert, nicht verändert) und können dann ins Webanno importiert werden.
Ist die Bearbeitung in Webanno abgeschlossen oder sollen die Daten zwischenzeitlich evaluiert werden, können sie exportiert und zurück ins Standardformat umgewandelt werden. Wichtig: Für eine vollständige Zusammenführung der alten und neuen Daten muss die Standardformatdatei von vor der Konversion ins Webanno-Format noch vorliegen, ansonsten gehen Metadaten verloren.
Die Daten im Standardformat können durch Applikationen durchsucht und ausgewertet werden.

2.2 Am Beispiel CNDH

Im Falle von CNDH liegen die Daten in einem HTML-Format vor mit grossen Mengen an unnötigen Informationen. Die relevanten Informationen werden mit dem HTMLToStandard-Skript in das Standardformat eingefügt.
Üblicherweise werden die Daten im nächsten Schritt getaggt. Dies geschieht mit dem Spanish_Tagger-Skript, welches einerseits eine erweiterte Datei im Standardformat ausgibt, wie auch eine importierbare Datei für Webanno anderer-

Tabelle 1: Struktur des Standardformat am Beispiel von getaggten CNDH-Daten

Stufe	Inhalt	Weitere Informationen
corpus	-	-
document	-	Name der Input-Datei, Suchwort
phrase	Originaler Satz	Stil, Komplettheit, Id, Metadaten, gefundenes Wort
token	Ein einzelnes Token des Satzes	POS, Lemma, Dependency Tag und Parent

seits. Manchmal werden für die Bearbeitung in Webanno keine Tags benötigt, dann kann auf das StandardToWebanno-Skript (in Arbeit) zurückgegriffen werden, welches ebenfalls eine importierbare Datei für Webanno zurückgibt.
Sind die Daten in Webanno bearbeitet worden, können sie mittels ToStandard zurück ins Standardformat konvertiert werden. Da Skript geht hierbei dynamisch mit selbst erstellen Layern um, es können also beliebig viele neue Layer in Webanno erstellt werden.
Zur Evaluierung steht im Moment nur das Search-Skript zur Verfügung, welches es erlaubt, über einen Suchsyntax passende Daten zu finden. Das genaue Format, in dem Suchergebnisse dargestellt werden, ist noch nicht festgelegt.

3 Benutzung der Applikationen

3.1 Allgemeine technische Voraussetzungen

- Python 3: Alle Applikation verwenden Python Version 3.5.2 als Programmiersprache. Es wird diese Version oder eine neuere benötigt. Sollte Python 3.5.2 oder neuer nicht installiert sein, kann es hier heruntergeladen werden.
- FreeLing: FreeLing ist eigentlich eine C-Bibliothek zur Verarbeitung von natürlicher Sprache, bietet aber Schnittstellen für Python. Vorteile sind die grossen Anpassungsmöglichkeiten und die grossen Ressourcen für spanische Sprache, die standardmässig enthalten sind. Nachteil ist die recht komplexe Installation, welche auch nur auf Linux und unter noch grösseren Schwierigkeiten auf iOS möglich ist. FreeLing wird nur für Skripte mit Tagging-Funktion verwendet. Download hier, Anleitung zur Installation für Python hier.
- Weitere Module: Für manche der Werkzeuge werden neben FreeLing die Python-Module lxml und regex benötigt.

3.2 Übersicht

Nach dem Start der Applikation werden dem Benutzer die möglichen Werkzeuge präsentiert. Sie werden im Folgenden im Detail erläutert.

3.3 HTML zu XML

3.3.1 Zweck

Das HTMLToStandard-Skript wird zur Konversion von CNDH-Daten aus dem HTML-Format in das StandardXML-Format verwendet. Ausgegeben werden eine StandardXML-Datei mit den Satzdaten, eine XML-Datei mit den Metadaten und eine zusätzliche Textdatei mit mehr Informationen zur Satztrennung.

3.3.2 Bedienung

Das Skript wird durch Doppelklick oder über die Kommandozeile geöffnet. In der obersten Spalte muss der Pfad zum Ordner gesetzt werden, in dem die zu bearbeitenden HTML-Dateien liegen. Es werden immer alle HTML-Dateien im Ordner geparst, daher sollten sich keine HTML-Dateien, die nicht bearbeitet werden sollen, im selben Ordner befinden.
In der zweiten Zeile muss ein Name für die neue StandardXML-Datei festgelegt werden, in der dritten ein Name für die Datei mit den Metadaten. Der Zusatz *.xml* muss nicht dazu angegeben werden (Beispiel: CNDHStandard, nicht CNDHStandard.xml). Dasselbe gult für die dritte Zeile. Ist alles eingetragen, kann der Prozess mit 'Konversion starten' angestossen werden.

Tabelle 2: Konversion von und zu verschiedenen Datenformaten bei CNDH

Von	Durch	Zu
HTML	HTMLToStandard	Standard-XML
Standard-XML	Spanish_Tagger	Standard-XML mit Tags
Standard-XML	Spanish_Tagger	Webanno-TSV mit Tags
Standard-XML	StandardToWebanno	Webanno-TSV
Webanno-TSV (mit Tags)	ToStandard	Standard-XML (mit Tags)
Standard-XML (mit Tags)	Search	Suchergebnisse

3.4 XML Taggen

3.4.1 Zweck

Mit dem Spanish_Tagger werden Standarddateien mit Tags versehen. Dazu wird ein statistischer POS-Tagger und ein statistischer Dependenzen-Parser benutzt. Ausgegeben wird sowohl eine getaggte StandardXML-Datei wie auch eine TSV-Datei, die in Webanno importiert werden kann.

3.4.2 Bedienung

Zuerst wird im ersten Feld der Pfad zur StandardXML-Datei, die es zu taggen gilt, eingegeben. Es kann auch der Computer mit einem Klick auf 'Datei wählen' durchsucht werden. Darunter muss ein Name für die resultierende, getaggte Datei eingegeben werden.
Nun kann noch festgelegt werden, ob bestimmte Sätze nicht bearbeitet werden sollen. In der resultierenden StandardXML-Datei werden diese Sätze weiterhin vorhanden sein, aber nicht mit Tags versehen.
Eine erste Filter stellt der Stil des Satzes dar. Im Moment existieren nur drei Optionen: 'all', 'poem' und 'plain'. 'all' schaltet diese Filter aus. 'poem' liest nur Sätze ein, die vom HTMLToStandard-Skript nicht als Gedichte gekennzeichnet wurde. Diese Kennzeichnung erfolgt über Merkmale wie Mehrzeiligkeit oder fehlende Satzzeichen. 'plain' liest umgekehrt alle Sätze ein, bei denen es sich nicht um Gedichte handelt.
Im nächsten Feld kann festgelegt werden, wie viele Sätze pro resultierender TSV-Datei geschrieben werden sollen. Hat man also 200 Sätze in einem Korpus und gibt hier '50' ein, erhält man vier Dateien zu 50 Sätzen. Diese Massnahme wurde eingeführt, da Webanno Probleme damit hat, sehr grosse Dateien zu importieren und exportieren.
Im Feld darunter kann ein Häkchen gesetzt werden, wenn nur komplette Sätze beachtet werden sollen. Ein kompletter Satz wird durch einen Grossbuchstaben oder ein beginnendes Satzzeiches zu Beginn und einem beendenden Satzzeichen zum Schluss definiert. Im letzten Feld schliesslich fällt der Entscheid, ob überhaupt eine TSV-Datei für Webanno ausgegeben werden soll, oder nur eine StandardXML-Datei gewollt ist. Mit 'XML zu Webanno' kann die TSV-Datei auch nachträglich noch erstellt werden.
Mit einem Klick auf 'Tagging-Prozess starten' wird der Prozess schliesslich begonnen. **Der Prozess dauert relativ lange! Solange der Prozess läuft, kann in der Applikation nichts angeklickt werden! Das Programm ist nicht abgestürzt, sondern es läuft!** Zum Schluss wird der Benutzer über den Abschluss des Prozesses informiert und die Daten sind nun zur Weiterverarbeitung bereit.

3.4.3 Tagsets

Bei dem Tagset für die Part-Of-Speech-Tags handelt es sich um das EAGLES-Set für Spanisch. Weitere Informationen sind hier zu finden:
`https://talp-upc.gitbooks.io/freeling-user-manual/content/tagsets/tagset-es.html`
Für die Dependenzen wird das Tagset des Ancora-Korpus verwendet:
`http://clic.ub.edu/corpus/webfm_send/49`
`http://clic.ub.edu/corpus/webfm_send/20`

3.5 XML zu Webanno

3.5.1 Zweck

Diese Applikation wandelt eine StandardXML-Datei in eine Webanno-Import-Datei um. Sie hält

sich dabei an dasselbe Schema wie der Tagger.

3.5.2 Bedienung

Die Bedienung ähnelt dabei sehr dem Tagger. Zuerst wird oben der Pfad zur StandardXML-Datei eingesetzt, falls nötig über den Knopf gesucht. Darunter sollte ein Name für die resultierenden TSV-Dateien gegeben werden. Nun muss noch angegeben werden, welche Stile erlaubt sind (wie beim Tagger), 'all' ist dabei der Standard. Dann muss festgelegt werden, wie viele Sätze pro TSV-Datei geschrieben werden sollen. Und schliesslich kann noch festgelegt werden, dass nur komplette Sätze verarbeitet werden sollen. Alle Sätze, welche die Filterbedingungen nicht erfüllen, werden nicht in die TSV-Dateien geschrieben. Wenn alles eingestellt ist, wird der Prozess mit dem Klick auf 'Starte Konversion' gestartet.

3.6 Webanno zu Standard

3.6.1 Zweck

Mit diesem Skript wird eine Datei aus Webanno zurück in ein StandardXML formatiert. Wenn eine originale StandardXML-Datei gegeben wird, werden die Tags zu den Sätzen hinzugefügt. Ohne originale StandardXML-Datei wird eine neue Datei erstellt, welche die Webanno-TSV-Datei in Standardformat darstellt. Wenn eine originale StandardXML-Datei gegeben wird, wird von der Applikation erwartet, dass Informationen zu den Metadaten gegeben sind in einer der Spalten des Webanno-TSVs. Diese Metadaten-Spalte sollte immer enthalten sein, wenn die Webanno-TSV-Datei ursprünglich mit dem Spanish-Tagger kreiert wurde. Sind die Metadaten nicht vorhanden, wird wie bei einer fehlenden StandardXML-Datei verfahren.

3.6.2 Bedienung

In die oberste Zeile wird der Pfad zum Webanno-TSV eingetragen. In der Zeile darunter, falls vorhanden, der Pfad zur originalen StandardXML-Datei. Zum Hinzufügen können drei verschiedene Modi gewählt werden:

- full overwrite: Alle bisherigen Tokens und Tags werden gelöscht und die Neuen angehängt. Hierbei handelt es sich auch um das standardmässige Vorgehen.
- replace changed ones: Hierbei handelt es sich nur um einen teilweisen Wechsel. Es werden nur solche Token geändert, die vorher auch schon existiert haben. Beispiel: In einer separaten Datei wurden nur Named Entities getaggt. Wir möchten diese Informationen mit den bisherigen zusammenführen (Mit der Datei, die alle anderen Informationen enthält wie POS, Lemmata, usw.). Für diese Methode werden ids benötigt, was bedeutet, dass auch die andere Datei zuvor durch Webanno gelaufen sein muss, bzw. daher auch durch dieses Skript (Da der Spanish-Tagger im Moment Tokens keine Idee gibt, das muss noch ergänzt werden).
- append: Es werden einfach alle Tokens des Webanno-TSV angehängt. Es erfolgt keine Überprüfung ob das Token schon vorhanden ist!

Die gegebene StandardXML-Datei wird dabei modifiziert, wenn man die Datei ohne Tags also behalten möchte, muss eine Kopie erstellt werden! Ist keine StandardXML-Datei gegeben, wird dem Namen der Webanno-TSV-Datei einfach noch ein 'Standard.xml' angehängt und die resultierende Datei so benannt. Mit 'Starte Konversion' wird der Prozess begonnen.

3.7 XML durchsuchen

3.7.1 Zweck

Diese Applikation dient zum Durchsuchen und Analysieren der Daten. Ausgegeben wird eine HTML-Datei mit den Suchergebnissen sowie eine XML-Datei, die nur auf die Sätze beschränkt ist, welche den Suchkriterien entsprechen.

3.7.2 Bedienung

Die Applikation besteht aus zwei Seiten, die Hauptseite und die 'Metadata'-Seite, in welcher weiteres Filtern ermöglicht.
Eine Suchanfrage stellen:

1. In der obersten Zeile wird der Pfad zur StandardXML-Datei gesetzt, die es zu Durchsuchen gilt. Darunter kann (optional) der Pfad

zu der Datei mit den dazugehörigen Metadaten gesetzt werden.

2. Nun gilt es die Tokens hinzuzufügen, die gesucht werden sollen, sowie die Beziehung zwischen den Token. Zuerst werden dazu die dritte bis zur siebten Zeile ausgefüllt. Die Zeilen können auch leergelassen werden, dann wird nach diesem Kriterium nicht gesucht. Es können also Tokens nach ihren Wortformen, Lemmaformen, Part-Of-Speech-Tags, Dependenzen-Tags oder selbstdefinierten Attributen gesucht werden. Sind alle Attribute definiert wie gewollt, kann das Token zur Suche hinzugefügt werden mit einem Klick auf '+Bedingung'.

3. Ab dem zweiten Token können Beziehungen zu bisher eingegebenen Token ausgesucht werden. Dazu klickt man auf das Menü 'No relation' und sucht dort die gewollte Beziehung aus und dann das Token, auf das sich die Beziehung bezieht auf dem Feld daneben. Mit einem '+Bedingung' wird das Token wiederum hinzugefügt.

4. Einen Spezialfall stellt der '+Option'-Knopf dar. Er kann an Stelle von '+Bedingung' verwendet werden und fügt das neue Token als eine Alternative des zuvor eingegeben hinzu.

5. Die Suche kann schliesslich mit einem Klick auf 'Suche' gestartet werden. Die Suche wird eine kurze Zeit dauern, dann öffnet sich eine Seite mit zutreffenden Sätzen von selbst (im Standardeditor des Benutzersystems).

6. Mit einem Klick auf 'Zurücksetzen' werden alle bisherigen Eingaben gelöscht.

Filtern nach Metadaten:

1. Es muss sichergestellt werden, dass ein Pfad zu einer entsprechenden Metadaten-Datei gegeben ist, ansonsten werden die Metadaten-Filterkriterien ignoriert.

2. Im Tab 'Metadata' können schliesslich die gewollten Einschränkungen eingegeben werden. Im Falle des oberen Fensters mit Autor-, Publikations-, Herkunfts- und Herausgeberinformationen wird, wenn kein Häkchen gesetzt wird, auch nach nicht genauen Treffern gesucht. Also würde 'Carlos', 'Sanchez' oder 'carlos sanchez' auch mit 'Carlos Sanchez' übereinstimmen. Ist das Häkchen hingegen gesetzt, würde nur noch genau 'Carlos Sanchez' mit 'Carlos Sanchez' als übereinstimmend gesehen.

4 In Planung

4.1 PRESEAAToStandard