Wofür eigentlich XML?

Es macht Sinn, der Frage nach den Vorzügen von XML-Schemata die Frage nach dem Mehrwert allgemeiner Textkodierung voranzustellen. Kodierungen folgen einer Systematik, die dem Text ein Format verleiht. Auch wenn erprobte Standards bestehen, ist das Wie im Grunde jedem Kodierenden selbst überlassen. Das Format kann Text strukturieren, ihn mit einer Funktion versehen oder besondere Merkmale hervorheben. Es ermöglicht zudem eine Anreicherung des Texts mit zusätzlichen (Meta-)Informationen, z.B. Kommentaren, Angaben zum Autor des Codes, zum Entstehungszeitraum, etc.. Durch die systematische Struktur kodierter Texte lassen sich einzelne Textbausteine und Metainformationen präzise ansteuern und abrufen. Kodierungsstandards erlauben zudem, dass Texte/ Codes auch maschinell ausgelesen und "verstanden" werden können (z.B. HTML). Dies gilt nicht weniger in den Geisteswissenschaften. Je systematischer die Daten kodiert sind, desto effizienter lassen sich diese durchsuchen und gesuchte Informationen abrufen. Dass der direkte Zugriff auf die kodierten Daten hier dennoch eher die Ausnahme darstellt, liegt nicht zuletzt daran, dass eine Vielzahl von Tools Benutzeroberflächen bzw. Suchmasken bereitstellen, die einfachere Suchanfragen und Operationen ermöglichen. Diese Masken funktionieren nur vor dem Hintergrund einer kodierten Textgrundlage, minimieren jedoch im Sinne einer Vermeidung von Komplexität den direkten Kontakt zwischen dem Code und dem Nutzer.

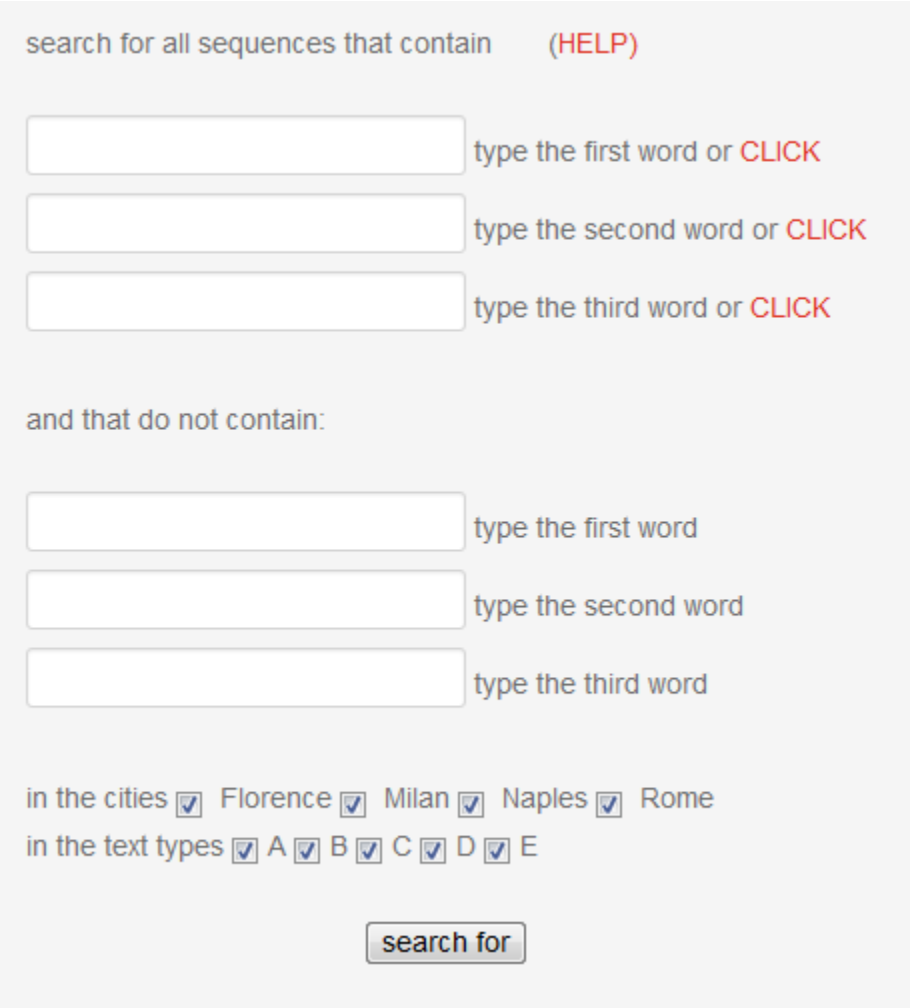
Was Masken (nicht) leisten können

Suchmasken bieten durchaus Vorteile. Sie fungieren als Übersetzer zwischen der Datengrundlage und dem Nutzer, indem sie einerseits den Input in eine komplexere Abfragesyntax umwandeln, andererseits mitgetragene Daten zugunsten einer nutzerfreundlicheren Lesbarkeit ausblenden. Dies ermöglicht einen intuitiveren Umgang mit In- und Output.

Literaturrechercheportale funktionieren beispielsweise auch hinreichend, ohne dass wir über ein Verständnis der Datenkodierung verfügen. Die Suchmaske genügt uns, um die gewünschten Inhalte zu finden. Das Kennen um das Kodierungsverfahren birgt somit keinen nennenswerten Vorteil.

Auch Korpora lassen sich meist anhand von Suchmasken durchsuchen. Für einfache Abfragen sind die vorprogrammierten Benutzeroberflächen völlig ausreichend. Mit steigender Komplexität der geplanten Anfragen wird man sich jedoch schnell der Begrenztheit der Möglichkeiten bewusst. Ein einfaches Beispiel soll dies veranschaulichen:

Nehmen wir an, wir möchten die Suchmaske des LIP-Korpus nutzen, um Artikelwörter zu finden.



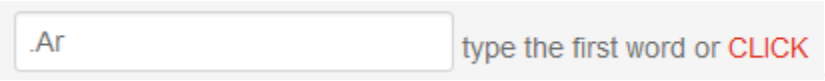
(Suchmaske des LIP-Korpus)

Die Oberfläche bietet uns hierfür mehrere mehr oder weniger geeignete Optionen. Zunächst lassen sich Artikelwörter über ihre Form definieren. Entweder wird hierzu jeder Artikel einzeln gesucht oder es werden Wildcards, sozusagen Platzhalter, verwendet, mit deren Hilfe sich ähnlich aussehende Artikel zusammenfassen lassen:

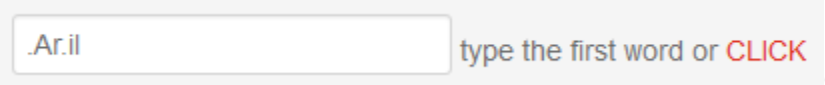
(Sucht nach "il")

(Sucht nach "l" gefolgt von einem beliebigen Zeichen)

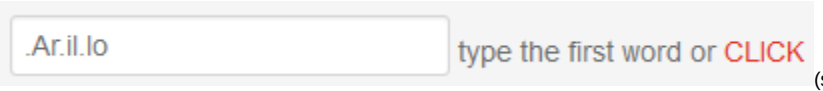
Abgesehen davon, dass diese Herangehensweise insbesondere bei allomorphen Strukturen einen Mehraufwand durch immer wieder neu zu startende Suchanfragen mit sich bringt, erschweren auch Homonyme die Suche. So sind die Formen */o* und */a* (bzw. /_) nicht ausschließlich der Kategorie der Artikel zuzuordnen sondern auch der der Pronomen (z.B. lo - bestimmter Artikel und direktes Objektpronomen). Alternativ lässt sich auch nach der Wortart – auch in Kombination mit dem Lemma oder mit dem Lemma und dem Token – suchen.



(Sucht nach der Wortart *.Ar*)



(Sucht nach dem Lemma *il*, welches der Kategorie *.Ar* angehört)



(sucht nach dem Token */o*, das als Lemma *il* besitzt und der Kategorie *.Ar* angehört)

Die beschriebenen Alternativen verraten bereits, dass die zugrundeliegenden Daten nicht als purer Fließtext vorliegen können, sondern in irgendeiner Weise mit Informationen zu Wortarten und Lemmata angereichert sind. Ein Blick in die XML-kodierten Daten bestätigt diese Annahme.

```
<parola>
  <word>
    il
  </word>
  <tag_g lemma="il" cg="Ar" />
</parola>
```

Auch wenn somit für jede Form eine abrufbare Annotation vorliegt, kann die Benutzeroberfläche die Suchmöglichkeiten, die eine solche Annotation bietet, nicht komplett nutzen. Möchte man zum Beispiel ausschließlich die Artikel */a* und */o*, d. h. /_, suchen, so kann dies nur ohne Angabe der Wortart erfolgen. Ein Input der Form *.Ar.il./_* erzeugt eine Fehlermeldung, während sowohl /_ als auch *.Ar.il./o* zulässig sind. Wie noch in dem weiteren Verlauf dieses Manuals deutlich werden wird, sind Abfragen annotierter Textmerkmale in unmaskierten Textdaten durchaus möglich.

Warum ist es also sinnvoll, sich mit Textdatenkodierung auszukennen? Zunächst schadet es nicht, sich damit auseinanderzusetzen, welche Möglichkeiten der Analyse bestimmter sprachlichen Daten überhaupt gegeben sind. Was wird eigentlich annotiert? Sind die Annotationen überhaupt linguistischer/ lexikalischer/ syntaktischer (etc.) Natur? Welche Annotationen nützen der jeweiligen Analyse?

Hat man sich mit der Datenstruktur auseinandergesetzt, so lassen sich Suchanfragen fernab der für die Suchmasken geltenden Grenzen formulieren. Um bei dem vorausgehenden Beispiel zu bleiben, ließe sich auf diese Weise nach Formen suchen, die aus zwei Buchstaben bestehen, deren zweiter Buchstabe einem *o* entspricht, deren Wortart mit einem *A* beginnt und deren Lemma nicht von Bedeutung ist (sozusagen *.A_.%._o*).

Darüber hinaus können Annotationen auch ergänzt oder modifiziert werden. Zum Beispiel lässt sich die Wortartangabe *Ar* in Kombination mit dem Lemma *il* mithilfe von regulären Ausdrücken in den gesamten Daten zu *bestAr* ändern (sozusagen *.Ar.il.%* > *.bestAr.il.%*).

Kodierte Textdaten folgen einer definierten Systematik, was sie auch für Maschinen lesbar macht. Ist das Datenschema, dessen sich ein Tool bedient, bekannt, lassen sich dem auch andere Textdaten anpassen. Auf diese Weise lässt sich zudem der Output eines Tools so abändern, dass er als Input eines anderen Tools verwendet werden kann.

Welchen Vorteil hat XML?

Die XML-Kodierung (eXtensible Markup Language) zeichnet sich insbesondere durch eine standardisierte und regelgeleitete Syntax aus. Es handelt sich um keine toolspezifische Auszeichnungssprache, sodass sie von jedem System gelesen werden kann. XML-kodierte Outputs diverser Tools lassen sich aufgrund der standardisierten Syntax leicht transformieren und anpassen. Daten im XML-Format sind zudem überaus nachhaltig, einerseits da sie für Computer stets lesbar bleiben, andererseits da auch mit geringem Aufwand zusätzliche Informationen annotiert werden können, ohne dass die Datenstruktur zerstört wird. Eine große Verbreitung des XML-Standards fördert zudem den wissenschaftlichen Austausch von Daten und Analysewerkzeugen.

Und dieses Manual?

Das hier startende Manual dient der systematischen Einführung in die Datenkodierung mittels XML. Es schafft Grundlagenkenntnisse, die dabei helfen sollen, XML-Strukturen zu erkennen und zu verstehen. Der sich anschließende 2. Block befasst sich mit der Praxis der XML-Kodierung. Hier werden nicht nur unterschiedliche Typen der Textkodierung beleuchtet, sondern auch der TEI-Standard vertieft. Block 3 widmet sich dem Nutzen von XML für sprachwissenschaftsrelevante Anwendungen. Im Zusammenhang mit ausgewählten Tools rückt hier auch die Möglichkeit der Datenmodifikation in den Vordergrund. Zuletzt befasst sich ein abschließender Block mit der Auswertung von XML-kodiertem Text mithilfe von regulären Ausdrücken und xPath.

Ausgewählten Blöcken werden Übungsaufgaben mit Lösungsvorschlägen beigefügt. Sie ermöglichen eine unmittelbare praktische Auseinandersetzung mit den XML-Strukturen und dienen der Routinisierung sinnvoller Operationen.

Up Datenkodierung und XML Ein erster Eindruck Next