Tutorial: Automatische Textannotation mit WebLicht

Inhalt

1. Was ist WebLicht? 1
2. WebLicht starten 1
3. Text importieren 2
4. Verarbeitungsketten 2
5. Wortarten-Tagging und Lemmatisierung 3
6. Herunterladen der Ergebnisse 3
7. Vorbereitung der WebLicht-Daten für die einfache Analyse mit Voyant 5

1. Was ist WebLicht?

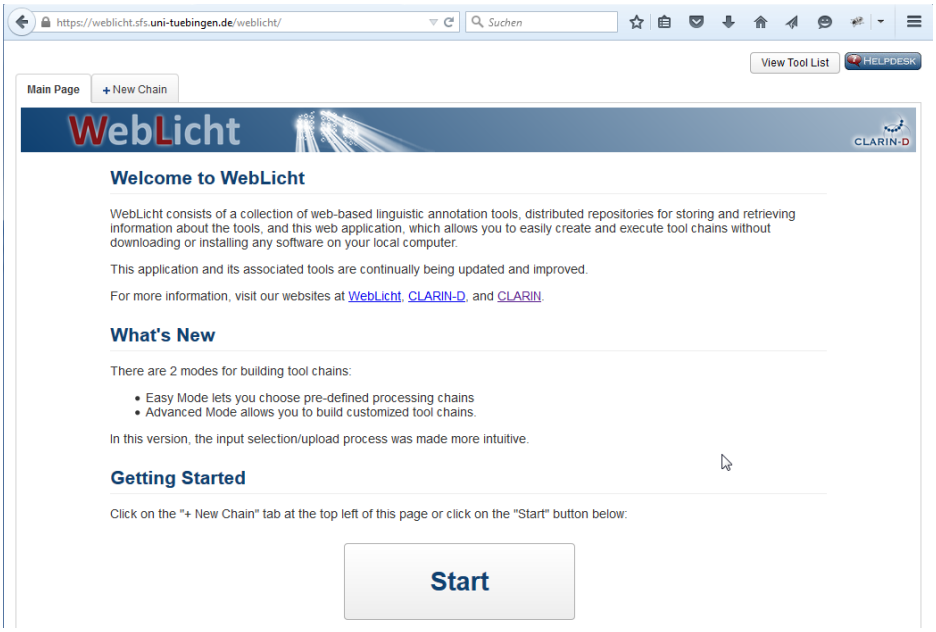
> WebLicht […] ist eine Service-orientierte Architektur (SOA) zur Erstellung annotierter Textcorpora. Sie wird seit Oktober 2008 […] entwickelt. Die Weiterentwicklung von WebLicht zu einer umfassenden virtuellen Forschungsumgebung stellt einen wichtigen Aspekt der Entwicklungsbemühungen innerhalb von CLARIN-D dar. Technisch wird WebLicht mittels Prozessketten von Restful Web Services umgesetzt. Jeder Web Service kapselt ein sprachtechnologisches Werkzeug, etwa die Abfragekomponente eines Korpus, einen Konverter, einen Tokenizer, einen Tagger, einen Parser oder dergleichen. Außerdem muss jeweils die Übersetzung von und zu den für das Werkzeug spezischen Ein- u. Ausgabeformaten geleistet werden. Jeder Web Service fügt mindestens eine Annotationsebene in Form spezifisch angereicherter Information hinzu. Am Ende steht ein auf verschiedenen Ebenen analysiertes Korpus, das in Form eines XML-Dokuments vorliegt. Damit die Web Services ineinandergreifen können, muss Kompatibilität zu einem von allen Diensten "verstandenen" gemeinsamen Austauschformat sichergestellt werden. Hierbei handelt es sich um das projektintern definierte Text Corpus Format (TCF). Letzteres ist weitgehend kompatibel mit bestehenden einschlägigen Formaten wie Negra, Paula, TüBa-D/Z etc., bzw. über spezifische Konverter jederzeit übersetzbar.
>
> **Quelle**: http://www.dig-hum.de/forschung/projekt/weblicht

2. WebLicht starten

Login

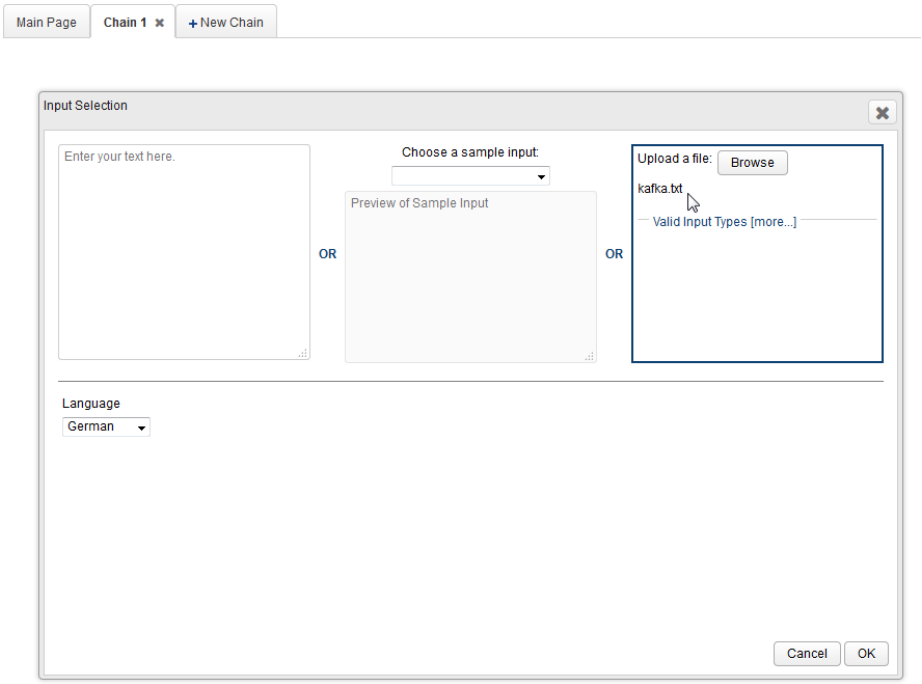
- Universität Regensburg auswählen
- Login mit NDS-Account und NDS-Passwort

https://weblicht.sfs.uni-tuebingen.de/weblicht/



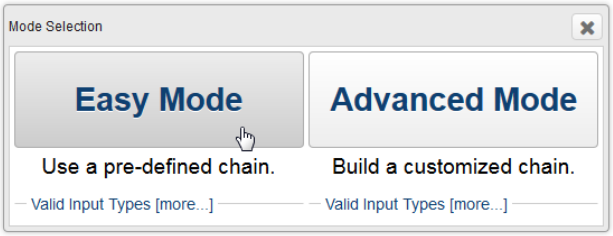
3. Text importieren

Zunächst muss der zu verarbeitende Text hochgeladen werden. Die Angabe der Sprache, in welcher der Text vorliegt, ist relevant für die Auswahl des entsprechenden Taggers, da bspw. ein englischer Tagger für deutsche Texte wenig sinnvolle Ergebnisse liefern würde.



4. Verarbeitungsketten

Bei WebLicht gibt es die Metapher der sog. „Chain“, also einer Kette einzelner Verarbeitungsschritte wie etwa Tokenisierung, Sentence Splitting, POS-Annotation, Lemmatisierung, u.v.m. Bei Bedarf können über „New Chain“ beliebige eigene Verarbeitungsketten definiert werden; für den Zweck dieser Übung soll aber die Standard-Verarbeitungskette (*Easy Mode*) ausreichen.



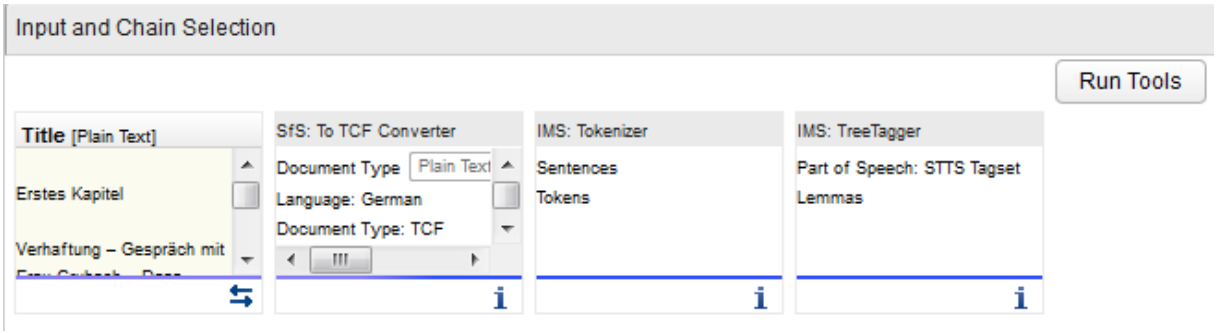
5. Wortarten-Tagging und Lemmatisierung

Nutzer haben nun die Wahl die Art der automatischen Annotation näher zu spezifizieren (Hinweis: es kann jeweils nur eine Option gewählt werden). Zur Auswahl stehen:

- Wortarten-Annotation und Lemmatisierung
- Morphologische Annotation
- Konstituenten-Parsing
- Dependenz-Parsing
- Annotation von *Named Entities*

Für diese Übung soll die Option „POS Tags/Lemmas“ ausgewählt werden. Die einzelnen Schritte der Verarbeitungskette sind am unteren Rand des Bildschirms aufgeführt: So wird zunächst der importierte Text in das einheitliche WebLicht-Format TCF (*text corpus format*) konvertiert (*To TCF Converter*). Als nächstes erfolgt eine automatische Segmentierung des Texts in Sätze und einzelne Tokens (*Tokenizer*). Schließlich erfolgt die automatische Analyse des Text durch den *TreeTagger*, welcher dann die entsprechend erkannten Wortarten und Grundformen für jeden einzelnen Token als Annotation zum Text hinzufügt.

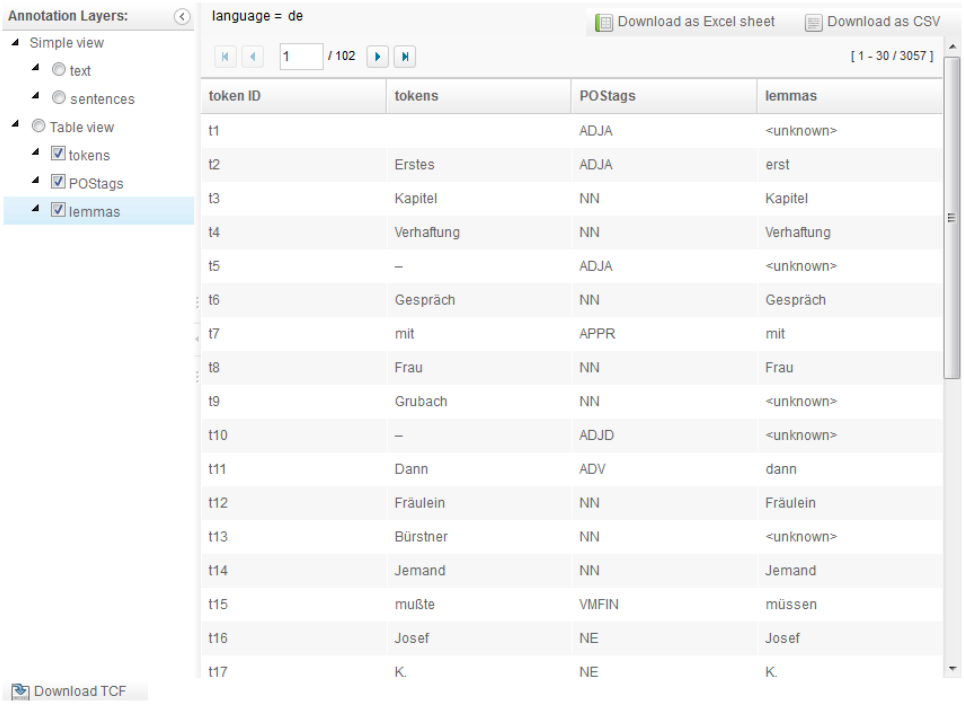
Die Verarbeitungskette wird mit Klick auf „Run Tools“ gestartet.



6. Herunterladen der Ergebnisse

Nach wenigen Sekunden werden die Ergebnisse des *TreeTaggers* im Browser angezeigt. Die Wortarten wurden nach dem Stuttgart Tübingen Tagset annotiert, eine vollständige Liste aller Wortarten-Tags finden Sie im Web[1].

[1] http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html



Zur weiteren Analyse können die Daten in unterschiedlichen Formaten heruntergeladen werden.

a) Zeilenweises, kommasepariertes Format → CSV (comma sepatared values)

Dieses Format kann in viele andere Formate konvertiert werden.

```
"token ID";"tokens";"POStags";"lemmas"
"t1";"";"ADJA";"<unknown>"
"t2";"Erstes";"ADJA";"erst"
"t3";"Kapitel";"NN";"Kapitel"
"t4";"Verhaftung";"NN";"Verhaftung"
"t5";"–";"ADJA";"<unknown>"
```

b) Tabellarisches Excel-Format

	A	B	C	D
1	token I	tokens	POStags	lemmas
2	t1		ADJA	<unknown>
3	t2	Erstes	ADJA	erst
4	t3	Kapitel	NN	Kapitel
5	t4	Verhaftung	NN	Verhaftung
6	t5	–	ADJA	<unknown>

c) XML-basiertes WebLicht-Format → TCF (*text corpus format*)

Das TCF ist nach den Prinzipien des *Linguistic Annotation Framework* (ISO-Standard) konzipiert und dadurch sehr flexibel einsetzbar. Das Format ist XML-basiert und nach dem Grundsatz eines sog.

Stand-off-Formats organisiert, d.h. die eigentliche Annotation ist physisch vom Primärtext getrennt und mit diesem über Referenzen verbunden.

Weitere Informationen zum TCF sowie auch Tools zur Darstellung von TCF-Daten finden sich hier:

http://weblicht.sfs.uni-tuebingen.de/englisch/tutorials/html/

7. Vorbereitung der WebLicht-Daten für die einfache Analyse mit Voyant

Die Speicherung von linguistisch annotierten Korpora in einem Stand-off-Format wie TCF hat zweifellos viele Vorteile, wirft aber für die unmittelbare Analyse mit Voyant einige technische Hürden auf, die zunächst mit einem einfachen Zwischenschritt umgangen werden sollen. Dabei sollen die WebLicht-Daten in ein XML-Format konvertiert werden, welches in Voyant mir einfachen XPath-Ausdrücken direkt analysiert werden kann. Zu diesem Zweck laden Sie die annotierten Daten von WebLicht im CSV-Format herunter.

Öffnen Sie die CSV-Datei in einem beliebigen Texteditor, und nehmen Sie die folgenden Änderungen am Dokument vor:

Schritt 1

Ersetzen Sie die erste Zeile …

```
"token ID";"tokens";"POStags";"lemmas"
```

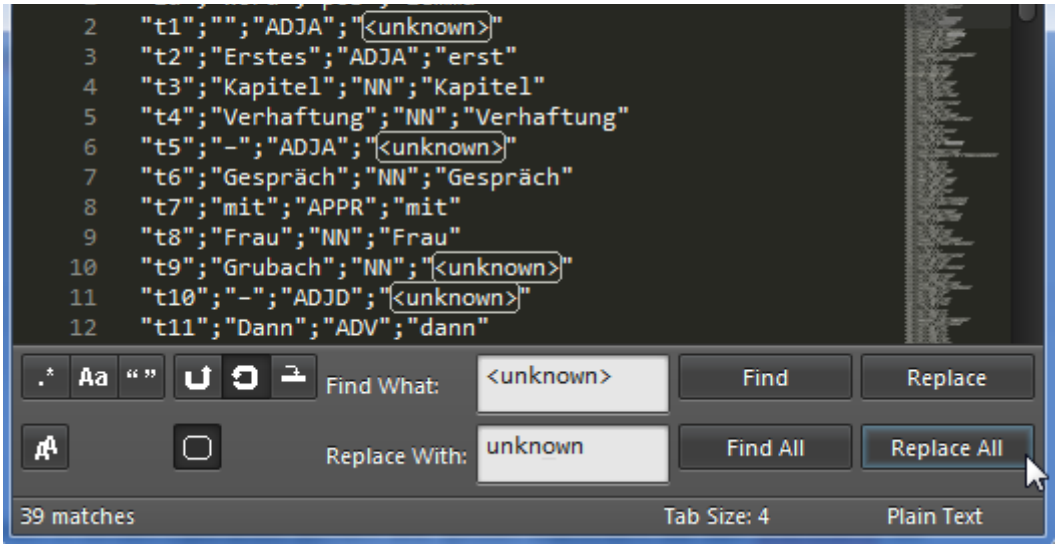
.. durch die folgende Zeile …

```
"id";"word";"pos";"lemma"
```

Erläuterung: in der ersten Zeile werden die Namen der XML-Tags spezifiziert, die im nächsten Schritt automatisch erstellt werden. Sie können hier im Wesentlichen beliebige Namen vergeben (keine Leerzeichen), sollten aber aus Konsistenzgründen zur Analyse-Übung zunächst die vorgeschlagenen Tag-Benennungen übernehmen.

Schritt 2

TreeTagger vergibt für jedes Wort bei dem kein Lemma gefunden werden konnte den Wert „<unknown>“. Bei der Transformation der Daten werden aus den spitzen Klammern jeweils Entitäten gebildet, d.h. „<unknown>“ wird dann „<unknown>“. Aus Gründen der leichteren Verarbeitung (es soll später nach *unknowns* gesucht werden) entfernen wird an dieser Stelle mit einer dokumentweiten Suchen-Ersetzen-Operation einfach alle spitzen Klammern bei den *unknowns*; im Texteditor *Sublime Text* würde dies etwa so aussehen:



Nach diesen beiden Schritten sollte ihre CSV-Datei so aussehen:

```
1  "id";"word";"pos";"lemma"
2  "t1";"";"ADJA";"unknown"
3  "t2";"Erstes";"ADJA";"erst"
4  "t3";"Kapitel";"NN";"Kapitel"
5  "t4";"Verhaftung";"NN";"Verhaftung"
6  "t5";"-";"ADJA";"unknown"
7  "t6";"Gespräch";"NN";"Gespräch"
8  "t7";"mit";"APPR";"mit"
9  "t8";"Frau";"NN";"Frau"
10 "t9";"Grubach";"NN";"unknown"
11 "t10";"-";"ADJD";"unknown"
12 "t11";"Dann";"ADV";"dann"
```

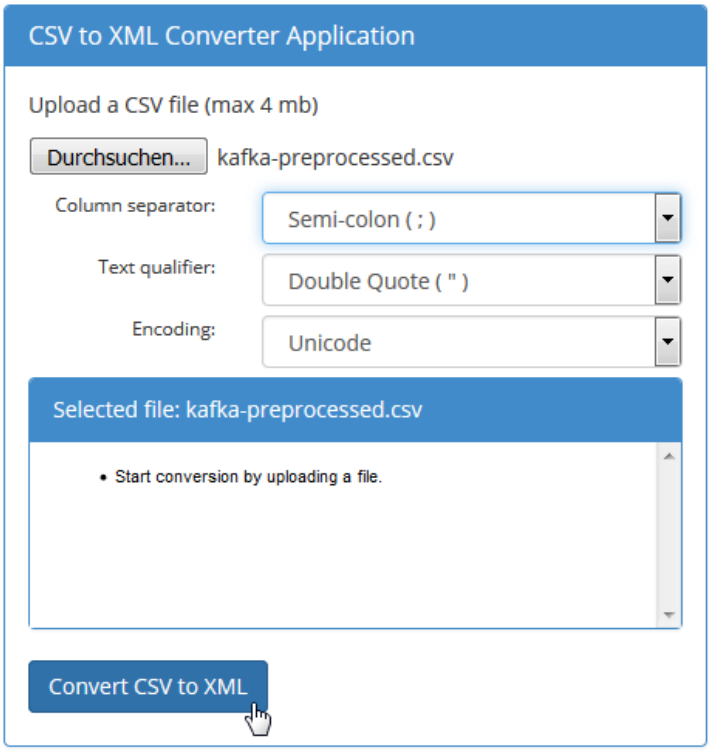
Speichern Sie die veränderte Datei unter einem beliebigen Namen, z.B. „kafka-preprocessed.csv“.

Schritt 3

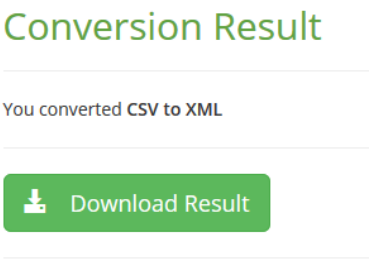
Im letzten Schritt können wir die derart vorbereiteten Daten schließlich über einen frei im Web verfügbaren *Converter* in ein einfaches XML-Format umwandeln. Navigieren Sie dazu auf die folgende Webseite:

`http://www.luxonsoftware.com/converter/csvtoxml`

Laden Sie die entsprechende CSV-Datei hoch, und wählen sie bei „*Column Separator*“ das Semikolon aus, da unsere Daten in jeder Zeile jeweils durch einen Strichpunkt getrennt sind.



Das Ergebnis können Sie im nächsten Schritt herunterladen, und in einem beliebigen Texteditor (oder Webbrowser) anzeigen lassen. Das Ergebnis der Konvertierung liegt zunächst als ZIP-Ordner vor, den Sie entsprechend entpacken müssen.



Die XML-Datei sollte so aussehen:

```
<Table1>
  <id>t6</id>
  <word>Gespräch</word>
  <pos>NN</pos>
  <lemma>Gespräch</lemma>
</Table1>
<Table1>
  <id>t7</id>
  <word>mit</word>
  <pos>APPR</pos>
  <lemma>mit</lemma>
</Table1>
<Table1>
  <id>t8</id>
  <word>Frau</word>
  <pos>NN</pos>
  <lemma>Frau</lemma>
</Table1>
<Table1>
  <id>t9</id>
  <word>Grubach</word>
  <pos>NN</pos>
  <lemma>unknown</lemma>
</Table1>
```

Sie können als letzten Schritt die Datei noch mit einem kürzeren Namen, z.B. „kafka-XML.xml“ abspeichern. Die so erstelle Datei kann nun z.B. mit Voyant weiter analysiert werden.