Crawling und Tokenisierung

Laden Sie mit dem Unixprogramm *wget* als Crawler Nachrichten-Seiten von `www.tagesschau.de` herunter.

Schreiben Sie dann ein Programm, welches aus den heruntergeladenen Seiten die reinen Texte der Zeitungs-Artikel ohne Werbung, Navigationselemente etc. extrahiert. Sie können hier die Python-Bibliotheken *re* (und/oder *html.parser*) verwenden. Verzeichnisse können Sie mit der Funktion *os.walk* durchwandern.

Aufruf: `python extract.py www.tagesschau.de > text.txt`

Schreiben Sie außerdem ein Tokenisierer-Programm, welches den gesamten extrahierten Text in Tokens (Wörter, Satzzeichen, Klammern etc.) zerlegt, und dann jeden Satz mit Leerzeichen zwischen den Tokens in einer separaten Zeile ausgibt.

Aufruf: `python tokenize.py abbreviations text.txt > text.tok`

Der Tokenisierer soll Abkürzungen korrekt behandeln. Eine Liste von deutschen Abkürzungen finden Sie hier:
`http://www.cis.uni-muenchen.de/~schmid/lehre/Experimente/data/abbreviations`

Zahlen (12.345) und Internetadressen (www.tagesschau.de) sollten nicht zerlegt werden. Schließende Klammern und Anführungszeichen nach einem Satzpunkt gehören noch zum Satz. HTML-Entities wie `&` sollten durch Unicode-Symbole ersetzt werden.

Versuchen Sie, kurzen und leicht verständlichen Code zu schreiben.

Vorüberlegungen

- Wie extrahieren Sie am besten die Texte aus den HTML-Dateien?
- Welche Schritte sind bei der Tokenisierung sinnvoll?

Sie dürfen Python-Bibliotheken, die nicht zum Standard gehören, (in dieser und anderen Übungen) nur verwenden, wenn sie explizit erlaubt wurden. Außerdem müssen Sie die Programme selbstverständlich selbst schreiben, wobei Gruppenarbeit von bis zu 3 Personen erlaubt ist.

Schicken Sie die beiden Programme an `schmid@cis.lmu.de`.

Überlegen Sie sich außerdem schon einmal, welche Sprache und welche morphologischen Phänomene Sie in der nächsten Übungsaufgabe behandeln wollen.