



Maschinelle Sprachverarbeitung

Übung

Aufgabe 1: Parsen und analysieren eines Nachrichtenkorpus

Mario Sängner

mario.saenger@informatik.hu-berlin.de

Korpus

- Parsen und analysieren des Reuters-Korpus
 - Beinhaltet eine Sammlung von englischsprachigen Nachrichtentexten
 - Zusätzliche Metadaten (z.B. Thema, Personen) vorhanden
 - Korpus liegt verteilt über mehrere XML-Dateien vor
- Der Korpus ist unter folgende URL verfügbar:
 - <https://box.hu-berlin.de/f/b6dfc700053541b3a21c/?dl=1>
 - Passwort wird per Email verteilt

XML-Struktur

```
<LEWIS>
  <REUTERS>
    <DATE>26-FEB-1987 16:20:06.60</DATE>
    <TOPICS> <D>money-supply</D> </TOPICS>
    <PLACES> <D>usa</D> </PLACES>
    <PEOPLE></PEOPLE>
    <ORGS></ORGS>
    <TEXT>
      <TITLE>NEW YORK BUSINESS LOANS FALL 195 MLN DLRS</TITLE>
      <DATELINE>NEW YORK, Feb 26 -</DATELINE>
      <BODY>Commercial and industrial loans on the books of
        the 10 major New York banks, excluding acceptances,
        fell 195 mln dlrs ...
      </BODY>
    </TEXT>
  </REUTERS>
  ...
</LEWIS>
```

Aufgabenstellung

- Implementierung eines **eigenen XML-Parsers**
 - SAX / DOM / Eigenentwicklung „from scratch“
 - Benutzung von Third-Party-Libraries zum Parsen von XML-Dateien (e.g. Jackson, Saxon) ist **nicht erlaubt**
 - Keine Validierung der XML-Struktur (mittels DTD) notwendig

Kennzahlen

- Anzahl an Dokumenten
- Anzahl von Wörtern in TITLE und BODY
 - Whitespaces (\s+) bilden Trennzeichen für Wörter
 - Alle Wörter in lowercase konvertieren
 - Anzahl an Wörtern insgesamt + Anzahl verschiedener Wörter („distinct“)
 - Keine Trennung zwischen TITLE und BODY!
- Die 30 häufigsten Wörter und deren Vorkommenshäufigkeit

Kennzahlen

- Anzahl / Vorkommen von folgenden Entitätstypen:
 - Topics (insgesamt und distinct)
 - Places (insgesamt und distinct)
 - People (insgesamt und distinct)

Abgabedetails

- Aufgabe muss mit Java oder einer Java-VM kompatiblen Sprache gelöst werden (z.B. Scala, Kotlin, ..)
- Programm lässt sich wie folgt starten:

```
java -jar uebung1-gruppeX.jar ordner/
```

- Programm liest und analysiert alle XML-Dateien (*.xml) im übergebenen Ordner
- Ausgabe der Ergebnisse in die Standardausgabe (stdout)

Programmausgabe

- Beispielhafte Ausgabe des Programms
 - Keine Musterlösung!

```
Anzahl Dokumente: 12
Anzahl Wörter: 1200 (178 distinct)
```

```
Anzahl Topics: 34 (22 distinct)
Anzahl Places: 41 (29 distinct)
Anzahl People: 29 (17 distinct)
```

```
Häufigste Wörter:
and          32
to           25
of           12
. . . .
```

Abgabe

- Upload eines ZIP-Archivs *uebung1-gruppeX.zip*
 - Ausführbares JAR-Archiv und Quellcode des Programms
 - Feedback zur benötigten Zeit
- Testet das Programm vor der Abgabe auf gruenau!
- Abgabe bis spätestens 13.11.2017, 23:59 Uhr über:
 - https://hu.berlin/ue_maschsp1718_ue1

Wettbewerb

- Implementiert eine schnelle (und korrekte!) Lösung
- Wir messen die Laufzeit des Programms mit "time"
 - Parallelisierung macht sicherlich Sinn
- Die drei besten Teams bekommen 3/2/1 Punkte!