Freitag: CQP

11 CQP
- Einleitung
- Indexierung eines eigenen Korpus
- Queries

CQP

Was ist CWB/CQP? I

- CWB[6]: Corpus Workbench, unterstützt Volltextsuche in lexikographischen und terminologischen Korpora
- read-only Format zur Speicherung von Korpora
  - mit Annotationen auf token-Ebene (POS, Lemmata, morphologische Merkmale, ...)
  - sowie flaches strukturelles Markup (Sätze, Chunks, Phrasen, ...).
- eigenes Korpus kann indexiert und komprimiert werden
- entwickelt vom IMS (Institut für Maschinelle Sprachverarbeitung, Stuttgart)

Was ist CWB/CQP? II

Hauptkomponenten von CWB:

- zentrale Komponente: CQP[7] - Corpus Query Processor: linguistische Suchmaschine
- Kommandozeilenprogramme[8] zur
  - Kodierung, Indexierung und Kompression von annotierten Textkorpora (`cwb-encode`, `cwb-makeall` u.a.)
  - Zugriff auf Häufigkeitsverteilungen im Korpus (`cwb-scan-corpus` u.a.)

---

[6] http://cwb.sourceforge.net

[7] `http://cwb.sourceforge.net/files/CQP_Tutorial/`

[8] `http://cwb.sourceforge.net/files/CWB_Encoding_Tutorial.pdf`

CQP

Indexierung des Korpus

CWB-Eingabeformat

- ein Wort je Zeile
- Spalten sind tab-getrennt
- eine Annotationsebene je Spalte (z.B. POS, Lemmata, ...)
- eine Spalte entspricht einem positionalen Attribut (**`p-attributes`**)
  default: **`word`**
- strukturelle Attribute (**`s-attributes`**) für die strukturelle Unterteilung des Korpus
  einfache, z.B. Sätze (**`<s>`**),
  oder mit Annotationen, z.B. Text-IDs (**`<text id='t23'>`**)

Beispiel: Korpus im CWB-Format

```
<corpus>
<text id='http://www.blabla.com'>
<s>
Regionality NN    regionality   unknown
always      RB    always        all
throws      VVZ   throw         verb.contact
up          RP    up            unknown
some        DT    some          unknown
interesting JJ    interesting   adj.all
quirks      NNS   quirk         noun.artifact
.           SENT  .             unknown
</s>
</text>
<text ...>
...
```

Kodierung eines Korpus

Umwandlung der Korpusdatei in Binärformat

```
cwb-encode -d <leeresVerzeichnis> -f  <Korpusdatei>
           -R <Registerverzeichnis>/<korpusname>
          (-P <positAttribut>){m}
          (-V <struktAttribut>){n}
          (-S <einfachesStruktAttr>){k}
```

-d Verzeichnis, in dem Korpusdaten gespeichert werden soll

-f Dateiname des Korpus

-R Zielname der Registerdatei (enthält Metaangaben) Wichtig: nur Kleinbuchstaben!

-P positionales Attribut, z.B. `-P pos` – Reihenfolge beachten!

-S einfaches strukturelles Attribut, z.B. `-S s` oder `-S np`

-V strukturelles Attribut, z.B. `-V text`

Kodierung eines Korpus

Umwandlung der Korpusdatei in Binärformat

```
cwb-encode -d <leeresVerzeichnis> -f  <Korpusdatei>
           -R <Registerverzeichnis>/<korpusname>
          (-P <positAttribut>){m}
          (-V <struktAttribut>){n}
          (-S <einfachesStruktAttr>){k}
```

- -d Verzeichnis, in dem Korpusdaten gespeichert werden soll
- -f Dateiname des Korpus
- -R Zielname der Registerdatei (enthält Metaangaben) Wichtig: nur Kleinbuchstaben!
- -P positionales Attribut, z.B. `-P pos` – Reihenfolge beachten!
- -S einfaches strukturelles Attribut, z.B. `-S s` oder `-S np`
- -V strukturelles Attribut, z.B. `-V text`

Kodierung eines Korpus

Umwandlung der Korpusdatei in Binärformat

```
cwb-encode -d <leeresVerzeichnis> -f  <Korpusdatei>
           -R <Registerverzeichnis>/<korpusname>
          (-P <positAttribut>){m}
          (-V <struktAttribut>){n}
          (-S <einfachesStruktAttr>){k}
```

- -d Verzeichnis, in dem Korpusdaten gespeichert werden soll
- -f Dateiname des Korpus
- -R Zielname der Registerdatei (enthält Metaangaben) Wichtig: nur Kleinbuchstaben!
- -P positionales Attribut, z.B. **`-P pos`** – Reihenfolge beachten!
- -S einfaches strukturelles Attribut, z.B. **`-S s`** oder **`-S np`**
- -V strukturelles Attribut, z.B. **`-V text`**

Beispiel: Registerdatei I

```
# corpus ID (must be lowercase in registry!)
ID   korpusname
# path to binary data files
HOME mycorpus

# corpus properties provide additional information
# about the corpus:
##:: charset  = "latin1" # change if your corpus
# uses different charset

## p-attributes (token annotations)
ATTRIBUTE word
ATTRIBUTE pos
ATTRIBUTE lemma
```

Beispiel: Registerdatei II

```
## s-attributes (structural markup)

# <text> ... </text>
STRUCTURE text                          # [annotations]

# <s> ... </s>
STRUCTURE s
```

Indexierung des Korpus

Erstellung des Lexikons und Indexierung

```
cwb-makeall -r <Registerverzeichnis>
            -V <KORPUSNAME>
```

KORPUSNAME ist Registerdatei, nun in upper-case

CQP

Starten von CQP[9]

```
cqp -e -r <Registerverzeichnis>
```

- e ⇒ input line editing
- r Registerverzeichnis
- weitere Optionen siehe **cqp -h**

cqp Prompt

```
:~$ [no corpus]>
```

[9] http://cwb.sourceforge.net/files/CQP_Tutorial.pdf

Nützliche Befehle (1)

- Anzeigen aller Korpora

```
show corpora;
```

- Arbeitsverzeichnis auf aktuelles Verzeichnis setzen
  (⇒ Speichern von Treffern)

```
set DataDirectory ".";
```

- Anzeigen aller Einstellungen

```
set;
```

  Ändern einer Einstellung, z.B.:

```
set LeftContext 5;
```

- Korpus KORPUSNAME aktivieren

```
KORPUSNAME;
```

Nützliche Befehle (2)

Nach Aktivierung eines Korpus

- Beschreibung des Korpus bzw. der Einstellungen anzeigen

```
show cd;
```

- Annotationsebene X mit anzeigen (z.B. X=pos)

```
show +X; (show +pos;)
```

Anfragen stellen (1)

- Einfachste Anfrage: Suche nach einem Wort, z.B. „Hut“

```
"Hut";
```

- Nach bestimmten Attributen suchen, z.B. attr

```
[attr="value"];
```

- Nach Wortsequenzen suchen, z.B. DT „Hut“

```
[pos="DT"] "Hut";
```

- Mehrere Beschränkungen für ein Wort

```
[pos="DT" & attr="value"];
```

- Einbeziehung von strukturellen Attributen

```
<np> "the" [lemma="house"];
```

- Beschränkung der Trefferanzahl

```
[lemma="house"] cut 2;
```

Anfragen (2): Reg. Ausdrücke

- ```
  [pos="V.*" & lemma="house"];
  [pos!="DT"];
  ```
- {n,m}: n bis m Vorkommen

  ```
  "dog" []{1,3} [lemma="bite"];
  ```
- Disjunktionen

  ```
  "[a-z]";
  "(he|she|it)";
  ```
  ```

Anfragen (3): Trefferanzeige

- Häufigkeitsverteilungen berechnen (Bezieht sich auf letzten Query)

```
count by pos;
```

- Sortieren nach Attribut

```
sort by lemma;
```

- Speichern der Treffer in Variable

```
var = [attr="bla"];
```

- Anzeigen der Treffer

```
cat var;
```

- Speichern in Textdatei

```
cat var > "dateiname.txt";
```

Übung 17