Tutorium der DGfS-Sektion Computerlinguistik

KORPUSLINGUISTIK MIT ONLINE-RESSOURCEN

EINE INTERAKTIVE EINFÜHRUNG FÜR LINGUISTEN

Reguläre Ausdrücke

Stefanie Dipper
Stefan Evert
Heike Zinsmeister

München, 28.01.2011

Suchmuster für Wörter

- Suche nach **Menge von Wörtern**, die einem bestimmten Muster folgen
  - Wörter, die auf *-ung* oder *-ungen* enden
  - Akronyme wie *E.S.S.T.* und *S.O.S.*
  - Wörter mit mehr als vier aufeinander folgenden Konsonanten u.ä.
  - Numeraladjektive wie *27-prozentig*, *5-fach*
  - Gibt es Wörter mit sechs *o*'s?

Reguläre Ausdrücke

- Beliebiges Zeichen: `.` (statt `?`)
- Suffix/Präfix: `.*ung` (statt `*ung`)
- Ein oder mehr Zeichen: `.+` (statt `+`)
- Alternative: `(auf|ab)` (statt `[auf,ab]`)
- Reguläre Ausdrücke sind kompositionell
  - komplexe Suchausdrücke entstehen duch Kombination elementarer Operatoren
- `(ha)+` ➔ *ha*, *haha*, *hahaha*, …

RA: Einzelne Zeichen

- Beliebiges Zeichen: `.`
- Metazeichen „wörtlich“: `\.`, `\?`, …
  - das Metazeichen wird durch `\` „geschützt“
- Zeichenauswahl: `[aeiou]`, `[a-z]`, …
  - Achtung: `[a-z]` schließt *ä*, *ö*, *ü*, *ß* nicht ein!
- Ausschluss von Zeichen: `[^0-9]`
  - alles außer Ziffern (*wirklich* alles!)

RA: Wiederholungsoperatoren

- Liste der Wiederholungsoperatoren
  - `(…)?` optional
  - `(…)*` beliebig viele Wiederholungen
  - `(…)+` eine oder mehr Wdh
  - `(…){n}` genau *n* Wiederholungen
  - `(…){n,m}` mindestens *n*, höchstens *m*
  - `(…){n,}` *n* oder mehr Wdh

RA: Wiederholungsoperatoren

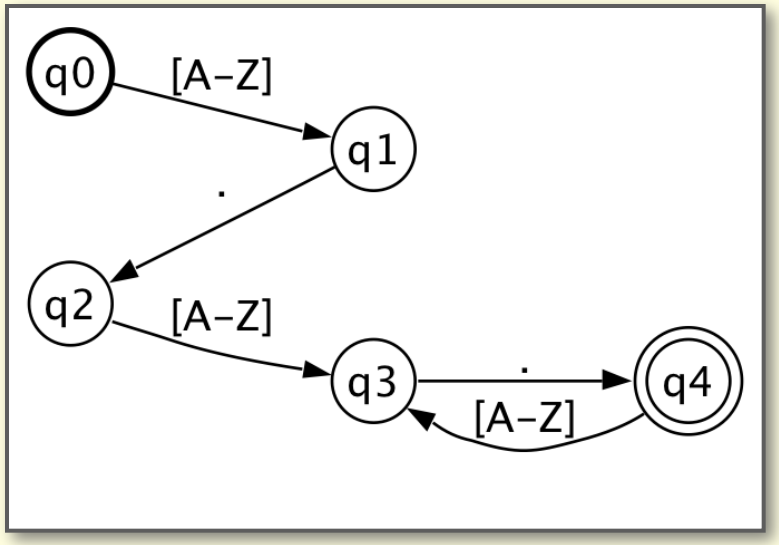
- Operatoren wirken auf …
  - `.+` 1 oder mehr beliebige Zeichen
  - `z+` *z*, *zz*, *zzz*, *zzzz*, *zzzzz*, …
  - `[0-9]+` 1 oder mehr Ziffern
  - `ha+` *ha*, *haa*, *haaa*, *haaaa*, …
  - `(ha)+` *ha*, *haha*, *hahaha*, …
  - `(ta|tü)+` *ta*, *tü*, *tatü*, *tatütata*, …
  - `(…)+` `(…)` kann komplexes Muster sein

RA: Beispiel

- Wir suchen Akronyme wie *S.O.S.*
  - zerlege Suchmuster in kleine Bestandteile
  - zwei oder mehr Wiederholungen von *A.*, *B.*, *C.*, *D.*, *E.*, … = `[A-Z]\.`
  - Wiederholungsoperator: `(…){2,}`
  - zusammen: `([A-Z]\.){2,}`
- Korpussuche (`EUROPARL-DE`): *I.F.*, *W.G.*, *S.A.*, *O.K.*, *U.S.*, *S.O.S.*, …

Vorteile regulärer Ausdrücke

- Für Anwender: komplexe Suchmuster mit wenigen Metazeichen
- Für den Computer: können reduziert werden auf Metazeichen `|`, `*` und `(...)`
- Implementierung mit endlichen Automaten (FSA) sehr effizient



Tutorium der DGfS-Sektion Computerlinguistik

KORPUSLINGUISTIK MIT ONLINE-RESSOURCEN

EINE INTERAKTIVE EINFÜHRUNG FÜR LINGUISTEN

CQP-Anfragesyntax

Stefanie Dipper
Stefan Evert
Heike Zinsmeister

München, 28.01.2011

CQP

- CQP ist der Corpus Query Processor der IMS Open Corpus Workbench (CWB)
  - schnelle Suche auf großen Textkorpora mit linguistischen Annotationen
- http://cwb.sourceforge.net/

CQP & reguläre Ausdrücke

- reguläre Ausdrücke auf Zeichenebene
  - `"([A-Z]\.){2,}"`
  - `"[0-9]+-[a-z]+" %cd` für Numeralkomp.
    - `%c` ignoriert Groß- und Kleinschreibung
    - `%d` findet auch Umlaute und Akzente
  - funktioniert nicht über Wortgrenzen hinaus!
- reguläre Ausdrücke auf Wortebene
  - z.B. PP = Prep `(`Det`)? ((`Adv`)?` Adj`)*` N

CQP & Tabellenformat

cpos	word	pos	lemma
0	Um	KOUI	um
1	den	ART	d
2	linguistischen	ADJA	linguistisch
3	Reichtum	NN	Reichtum
4	zu	PTKZU	zu
5	beweisen	VVINF	beweisen
6	,	$,	,
7	welchen	PRELS	welch
…	…	…	…
46	.	$.	.

CQP-Syntax

- Tokenmuster `[…]` ➔ Tabellenzeilen
  - Zugriff auf beliebige Annotationen:
    `[pos = "VV.*"]`, `[lemma = ".*ung"]`
  - `"[0-9]+"` kurz für `[word = "[0-9]+"]`
  - logische Konnektoren (Boolsche Ausdrücke)
    `&` (und), `|` (oder), `!` (nicht), `!=` (trifft nicht zu)
  - z.B. `[lemma = "unter.*" & pos = "VV.*"]`
  - auch direkter Vergleich: `[lemma != word]`

CQP-Syntax

- Reguläre Ausdrücke über Tokenmuster
  - `[…]` entspricht Zeichen(auswahl),
    `[]` entspricht `.` („matchall“)
  - Wiederholungsoperatoren: `?`, `*`, `+`, `{m,n}`,
    Alternativen `(…|…|…)` mit Schachtelung
  - Beispiel: einfache NP mit Kopf auf *-ung*
  - ```
    [pos = "ART"]? [pos = "ADJA"]*
    [pos = "NN" & lemma = ".+ung"]
    ```
    ```