

2. Grundmöglichkeiten von CQP:

Die prinzipielle Aufgabe des CQP ist die, Möglichkeit zu bieten, eine Korpusanfrage so genau wie möglich auszudrücken, damit die Anfrage zu einer möglichst kleinen Menge spezieller, aber für das Problem relevanter Belege führt, anstatt zu einer großen Menge von Konkordanzen, die nachträglich manuell inspiziert und zusätzlich gefiltert werden müssen.

Wichtig hierfür sind die möglichen Attribute der Korpusannotationen, nach denen der CQP entsprechend aufbereitete Korpora nach folgenden Attributen durchsuchen kann. Die „wichtigsten“ Attribute sind:

- Das Standard-Attribut *word*:
- Weitere Attribute:
 - Position
 - Wortklasseninformation
 - Lemmata
 - Morphosyntaktische Annotation
 - Strukturelle Attribute

Die Abfragesyntax differiert **stark** nach den jeweiligen Korpusannotationen. Lediglich das Standard-Attribut *word* kann mit jedem Korpus verwendet werden.

3. Zusatz-Funktionen von XKWIC gegenüber „reinem“ CQP:

- Das Anfrageergebnis wird in einer KWIC-Konkordanz präsentiert, wobei der angezeigte Ausschnitt über *Scrollbars* verändert werden kann.
- Das präsentierte Anfrageergebnis kann nach unterschiedlichen Kriterien sortiert werden.
- Durch Selektion einer einzelnen Konkordanzzeile wird diese mit einem erweiterten, benutzerdefinierbaren Kontext angezeigt.
- Das Anfrageergebnis oder ein selektierbarer Teil davon kann in einer Datei textuell abgespeichert werden.
- Einzelne Konkordanzzeilen können manuell aus dem Anfrageergebnis gelöscht werden oder es können neue Ergebnismengen manuell durch Kopieren aus anderen Ergebnismengen erzeugt werden.
- Eine einfache *Query-History* wird mitgeschrieben, diese besteht aus einer Liste, in der Anfragetext und Name des zugrundeliegenden Subkorpus festgehalten wird. Diese Liste kann gespeichert werden und so in nachfolgenden Sitzungen wieder zur Verfügung stehen.

4. Aufbau von XKWIC:

- Menüzeile (1)
- Suchanfragen-Zeile (2)
- Korpusauswahl-Zeile (3)
- Statuszeile (4)
- Ausgabefenster (5)
- Kontextfenster (6)



4.1. Menüzeile:

Hier werden Optionen für die Suche und die Erstellung und Bearbeitung von Subkorpora eingerichtet. Folgende Menüs sind vorhanden:

File: <ul style="list-style-type: none">- Options:- Display Attributes:- Credits: Copyright-Meldung- Quit	Concordance: <ul style="list-style-type: none">- Delete- Write to file- Print- Copy to subcorpus- Divide into subcorpora- Sort Concordance- Reduce Concordance- Set Collocate- Select/Unselect- Frequency Distributions
Query History: <ul style="list-style-type: none">- View- Save to file- Load from File- Clear	Subcorpora <ul style="list-style-type: none">- Save all Subcorpora
Windows: <ul style="list-style-type: none">- Alignment Window- Extended View- Messages	Help: <ul style="list-style-type: none">- Help (am SfS nicht verfügbar)- E-Mails an die Korpusadministratoren.- E-Mails an die Entwickler.

4.2. Suchanfragenzeile:

Hier wird die Suchanfrage eingegeben und mit dem [Start Query]-Knopf gestartet.

4.3. Korpusauswahl-Zeile:

Durch Betätigung des [?]-Knopfes wird die Liste der verfügbaren Korpora und Subkorpora geöffnet, das gewünschte Korpus kann mit der Maus ausgewählt werden. Es ist über einen Optionsschalter auch möglich, nur die Subkorpora anzeigen zu lassen.

4.4. Statuszeile:

Hier werden Nachrichten wie Suchergebnisse, Fehlermeldungen oder aktiver Korpus angezeigt.

4.5. Ausgabefenster:

Hier werden die Suchergebnisse angezeigt, diese können mit der Maus ausgewählt und weiterverarbeitet werden.

4.6. Kontextfenster:

Hier wird das zuletzt ausgewählte Suchergebnis in erweiterem Kontext angezeigt. Die Kontextgröße ist über [File] [Options] einstellbar oder kann direkt in die Suchanfrage eingebaut werden.

5. Anfragesyntax:

- **Generelles:** Eine Anfrage wird mit einem **Semikolon** bestätigt! Attribute (word, pos, lemma,...) werden in **Eckklammern** gesetzt.

```
[word="beispiel1"];
```

- **Wortattribute** können auch ohne Attributsangabe gesucht werden (Standardeinstellung):

```
[word="beispiel1"];
```

oder

```
"beispiel1";
```

Wichtig: Leerzeichen gelten als Zeichen!

- **Reguläre Ausdrücke** / Bool'sche Operatoren:

Disjunktionsoperator Bar '|':

"sucht|Sucht"; findet sucht in Groß- oder Kleinschreibung

Konjunktionsoperator Et '&'

[word="Regen" & pos="N.*"]; findet das Wort 'Regen' als Nomen.

Negationsoperator Minus '-'

[word="Regen" & pos!="N.*"]; findet das Wort Regen als Nicht-Nomen (Regen sich im Frühling die Schmetterlinge wieder?)

- **Platzhalteoperatoren**

Der **Operator Punkt** '.' entspricht einem beliebigen Zeichen und der **Operator Asterisk** '*' entspricht einer beliebigen Länge des vorangehenden Zeichens.

"lauf.*"; findet daher alle Formen wie ,lauf', ,laufen', ,laufe', ,laufend',...

"lauf*"; würde hingegen Formen wie ,lauff' oder ,lauffffff' finden.

Der **Operator Plus** '+' stellt eine Sonderfunktion des Operators Asterisk '*' dar: Während der Asterisk auch einen **Nullwert** repräsentieren kann, muss Plus eine **positive Zahl** darstellen.

"lauf.*"; findet auch ,lauf'

"lauf.+"; findet nur ,laufe', ,laufen',...

Da sich Asterisk und Plus aber nur nach dem vorangehenden Zeichen orientieren (ober Punkt), gilt auch:

"a+"; findet ,a', ,aa', ,aaa', ...

Unspezifizierter Platzhalter:

[]; findet alle Positionen im Korpus, es kann besonders als Platzhalter zwischen entfernt positionierten Suchbegriffen genutzt werden.

"gebe" []* "auf"; findet '...gebe das Fischereigeschäft zu Gunsten meines Neffen auf.', aber auch '...gebe nichts auf die Sprüche von unserem Bundesfinanzminister.'

- **Gruppierungen:**

Gruppierungsoperatoren sind Klammern '(' ')'. Der Verkettungsoperator (nicht sichtbar, z.B. "ab"; verkettet "a" und "b") dominiert über die Bool'schen Operatoren.

"ab|c"; findet 'ab' oder 'c'

"a(b|c)"; findet 'ab' oder 'ac'

- **Optionale Auslassungen:**

Der **Auslassungsoperator Fragezeichen** '?' läßt eine Suche nach optionalem vorangehendem Zeichen zu:

"suchen?"; findet 'suchen' oder 'suche'. (= "suche|suchen|sucht";)

- **Sonderzeichen:**

Sonderzeichen werden entweder im Zeichencode des Zeichensatzes (schwieriger) oder im Latex-Format (einfacher) ausgedrückt:

"Sp\344tzle" findet 'Spätzle' (Code)

"Sp\ "atzle" findet 'Spätzle' (Latex)

Auch die **von Operatoren belegten Zeichen** werden durch Voranstellen eines ‚Backslashes‘ als Zeichen markiert (Latex-Format):

"\" * " ; findet *

"\" . " ; findet .

"\" + " ; findet +

"\" | " ; findet |

"\" (" ; findet (

"\") " ; findet)

- Morphosyntaktische Attribute

Die Annotationen sind korpusabhängig. Daher müssen bei jedem Korpus über den [?]-Knopf und die [Info]-Funktion die jeweiligen Tagsets nachgesehen werden.

Beispiele können sein:

[pos="ADV"] ;

[pos="ADJ"] ;

[pos="N. * "] ;

(Beispiele aus dem MK parsed).

Lemmatisierte Korpora können mit folgendem Befehl nach Lemmata durchsucht werden:

[lemma="sehen"] ;

- Multiplikatoren:

X{n} bedeutet genau n Wiederholungen von X

X{n, } bedeutet n oder mehr Wiederholungen von X

X{n,m} bedeutet n bis m Wiederholungen von X

Beispiele:

"gebe" [] {3,7} "auf" ; oder komplexer:

([lemma="suchen"] [] {0,5} [word="zu"]) {0,3} [pos="INF"] ;

Eine weitere Variation ist der within-Parameter: Er bestimmt innerhalb wievieler Positionen sich das Ergebnis befinden soll:

"gebe" []* "auf" within 7 ;

"gebe" []* "auf" within 7 word ;

"gebe" []* "auf" within 7 words ;

sind drei Variationen, die dem Befehl "gebe" [] {0,5} "auf" ; entsprechen.

Multiplikatoren können auch dazu dienen, Kontext anzuzeigen:

[]*{5} "Mineral\ "olsteuer" []{5} ; zeigt das Wort 'Mineralölsteuer' mit einem Kontext von jeweil 5 Worten vor und nach dem Wort.

6. Anfragebeispiele:

- genese in Groß- oder Kleinschreibung:
"[gG]enese";
- ein mit „verwirr“ beginnendes Wort, das von einer Präposition oder einem Personalpronomen gefolgt wird:
"verwirr.*" [pos="IN" | pos="PP"];
oder:
"verwirr.*" [pos="IN|PP"];
- dasselbe, aber mit höchstens 10 Worten zwischen den Suchobjekten:
"verwirr.*" [] {0,10} [pos="IN" | pos="PP"]
- dasselbe, aber ohne Punkt zwischen den Suchobjekten:
„verwirr.*" [word!="\."] {0,10} [pos="IN" | pos="PP"]
- Ein Satz mit einem Adjektiv, einem Nomen, einer Konjunktion und einem weiteren Nomen:
[pos="JJ.*"] [pos="N.*"] "und|oder" [pos="N.*"]

7. Aufrufen von XKWIC am Sfs über einen beliebigen Unix/Linux-Rechner oder ein Terminalfenster:

```
xhost penthesilea.sfs.nphil.uni-tuebingen.de
ssh -l Benutzername penthesilea.sfs.nphil.uni-tuebingen.de
passwort
setenv DISPLAY=Rechnername.uni-tuebingen.de:0
Xkwic &
```

9. Eine kleine Aufgabe:

Wie lautet die kürzeste Suchanfrage, mit der man alle Formen von *treffen* findet?

```
"[tT]reff(e(nd)?|s?t)?|[tT]riff(s?t)?|[tT]raf(s?t|en)?|[tT]r\"af(en?|s?t)?
|getroffen";
```

8. Umfassende Literatur:

Christ, Oliver et al.: The IMS Corpus Workbench. Corpus Query Processor.
"http://www.ims.uni-stuttgart.de/CorpusWorkbench/HTML/" Version 2.2. (2.08.99).