Korpuslinguistik

Annis$_2$-Korpussuchtool

Suchen in tief annotierten Korpora

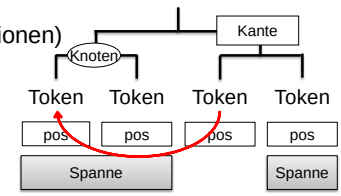
SE Historische Korpora, 26.06.2012
Prof. Anke Lüdeling, Carolin Odebrecht
Präsentation: Malte Belz, malte.belz@hu-berlin.de

Überblick

- Was ist Annis?
- Wie sieht das Webinterface aus?
- Wie werden Suchanfragen formuliert?

Was ist Annis?

- **ANNIS** steht für
  - **ANN**otation of **I**nformation **S**tructure
  - http://www.sfb632.uni-potsdam.de/d1/annis/
- Suchmaschine für tief annotierte, multimodale Korpora
  - Token (-annotationen)
  - Spannen
  - Bäume
  - Pointer



Links

- **Annis-Portal (öffentlich)**
  http://korpling.german.hu-berlin.de/Annis/search.html
- **Annis-Portal speziell für Lernerkorpora (Falko)**
  http://korpling.german.hu-berlin.de/falko-suche

Jetzt: Übung mit

- **Annis-Portal für diachrone Korpora (DDD)**
  http://korpling.german.hu-berlin.de/ddd/search.html

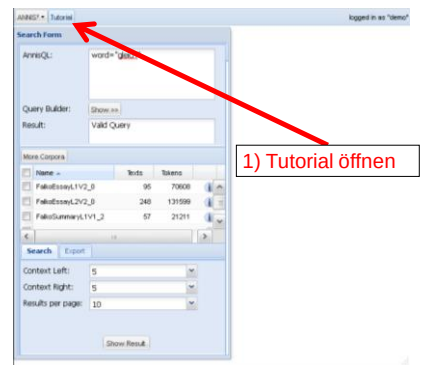
Bitte jetzt diesem Link folgen

Ziele der heutigen Sitzung

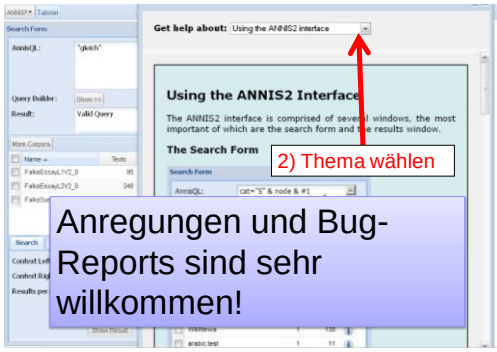
http://korpling.german.hu-berlin.de/ddd/search.html

- Wie und was kann man in ANNIS suchen?
  - Linguistische Muster
  - Token-Annotationen (Lemmata/Wortarten)
  - Spannenannotationen
  - Syntaktische Annotationen
    - Konstituenten
    - Dependenzen
  - Annis findet nur das, was annotiert ist!!!
- Wie sucht man nach mehreren/beliebigen Annotationen gleichzeitig?
- Wie filtert man nach Metadaten?

Das Web-Interface: Tutorial



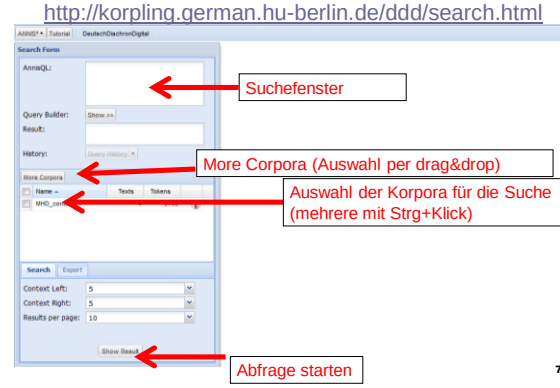
Das Web-Interface:Tutorial



6

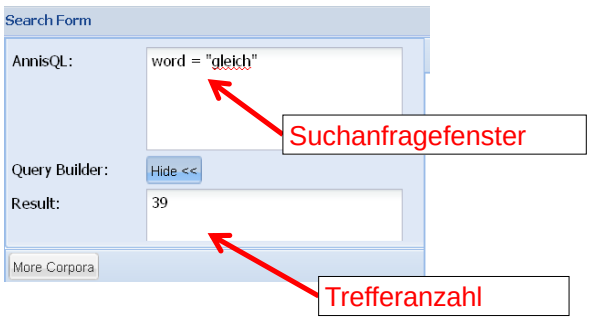
Das Web-Interface: Abfrage

http://korpling.german.hu-berlin.de/ddd/search.html



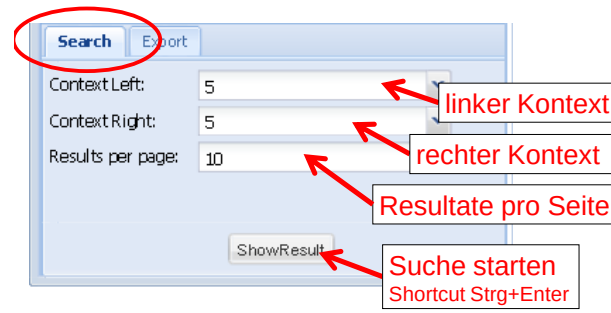
7

Das Web-Interface: Suchfenster



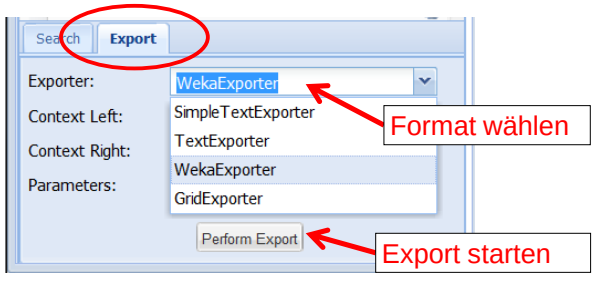
8

Das Web-Interface: Such-Einstellungen



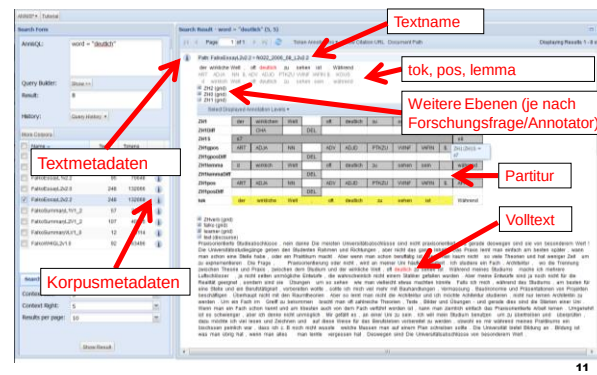
9

Das Web-Interface: Export-Einstellungen



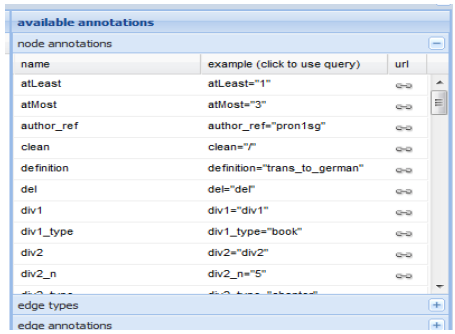
10

Das Web-Interface:Treffer



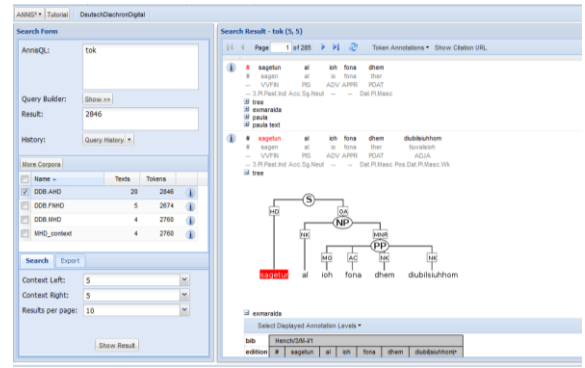
11

Web-Interface: Meta-Information



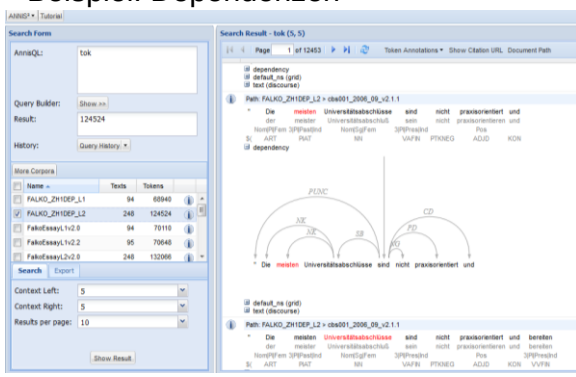
12

Beispiel: Syntaktische Beziehungen



13

Beispiel: Dependenzen



14

Wie suchen wir?

- Prinzip I: Variablen-Wert-Paare (Attribut-Wert-Paare, Layer-Wert-Paare)
- Prinzip II: Relationssuche

15

Prinzip I: Variablen-Wert-Paare

- tok = "das"

Anfrage in Anführungszeichen: findet exakt diesen String!

Variable (Layer) **Wert**

tok	Sofern	**das**	System	herrscht
pos	KOUS	ART	NN	VVFIN
lemma	sofern	d	System	herrschen

16

Prinzip I: Variablen-Wert-Paare

- pos = "NN"

Variable **Wert**

tok	Sofern	das	System	herrscht
pos	KOUS	ART	NN	VVFIN
lemma	sofern	d	System	herrschen

17

Prinzip I: Variablen-Wert-Paare

- lemma = "d"

Variable3 ("Lemma")

Wert

…findet *die, dem, den, …*

word	Sofern	das	System	herrscht
pos	KOUS	**ART**	NN	VVFIN
lemma	sofern	d	System	herrschen

18

beliebig erweiterbar…

- satz = "NS"

Variable4 ("Satztyp")

Wert

word	Sofern	das	System	herrscht
pos	KOUS	ART	NN	VVFIN
lemma	sofern	d	System	herrschen
satz	**NS**			

19

beliebig erweiterbar…

- satz = "NS"

…findet alle Nebensätze wie
Sofern das System herrscht

(sofern die Daten wie gezeigt annotiert sind)

20

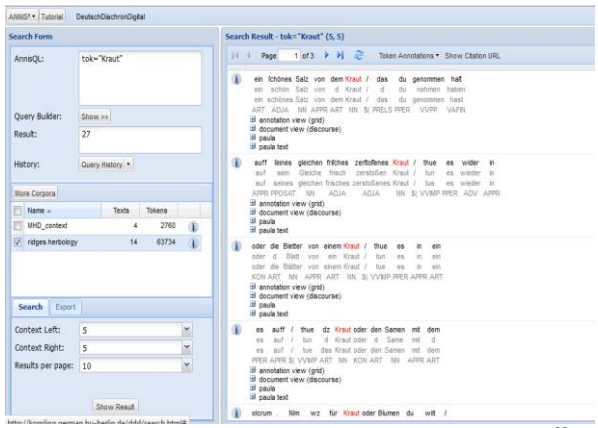
Aufgabe 1

http://korpling.german.hu-berlin.de/ddd/search.html

- **Laden des Korpus „ridges.herbology“ (more corpora→ drag&drop→check)**
- **Suchen Sie nach allen Vorkommen des Tokens „Kraut“**

tok = "Kraut"

21



22

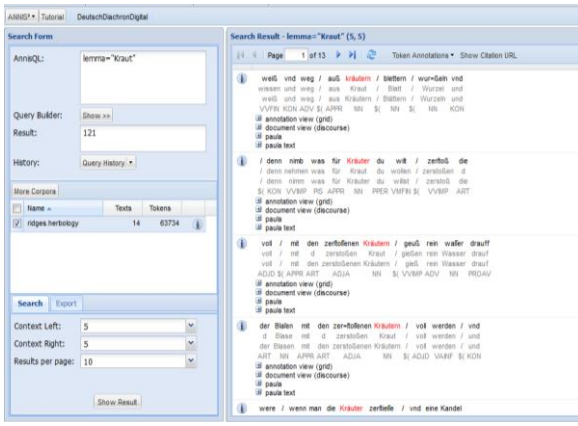
Aufgabe 2

- **Suchen Sie nach allen Vorkommen des Lemmas „Kraut“**

lemma = "Kraut"

☐ Ergebnis: 121

23



24

Mustersuche (reguläre Ausdrücke)

- Annis$_2$ erlaubt Mustersuchen auf allen Annotationsebenen
- Mustersuchen werden statt in " " in / / eingefügt
- Z. B. kann man damit nach allen Wörtern suchen, die „*kraut*“ enthalten.

```
tok = /.*kraut.*/
```

25

Operatoren: Joker .

- . ein beliebiges Zeichen al. → *als, alt, ...*
- . . zwei beliebige Zeichen al.. → *alle, alte, also*
- . . . drei beliebige Zeichen al... → *alles, altes, alias, ...*

Aufgabe 4

- **Welche Wortformen erhalten Sie?**

```
tok = /g.b./
```

Lösung: *gibt, gebe* (in ridges.herbology).
Denkbar wäre auch *gäbe*

27

Operatoren: ? und * +

das? das vorherige Zeichen ist optional
→ ϕ, s → *da, das*

das* das vorh. Zeichen kommt 0- bis ∞mal vor
→ ϕ, s, ss,… → *da, das, dass, dasssss*

das+ das vorh. Zeichen kommt 1- bis ∞mal vor
→ s, ss,… → *das, dass, dassssss*

28

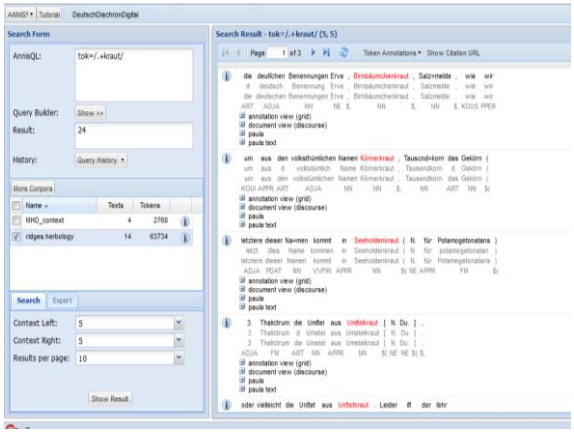
Beispiel 5

- **Versuchen Sie alle Token zu finden, die auf *kraut* enden.**

```
tok = /.+kraut/
```

Findet *Birnbäumchenkraut, Körnerkraut, Wielandskraut...*

29



30

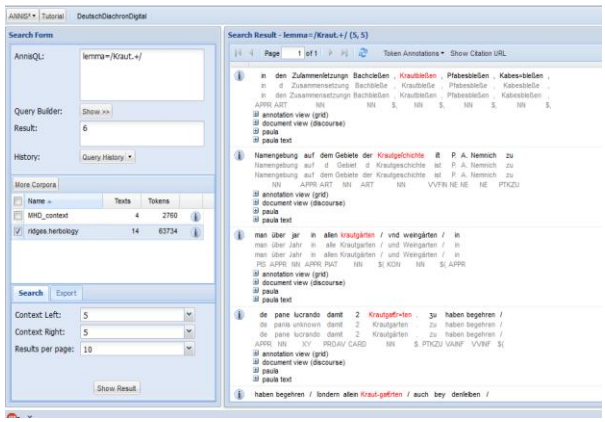
Beispiel 6

- **Versuchen Sie alle Lemmata zu finden, die mit *Kraut* beginnen.**

```
lemma = /Kraut.+/
```

Besonderheiten der Lemma-Ebene:
- *Kraut* wird zu Beginn großgeschrieben (keine Alternativensuche notwendig
- Es kann passieren, dass der Tagger das Lemma nicht kennt und mit <unknown> annotiert hat, falls diese Ebene nicht manuell korrigiert wurde

31



32

Alternativen: a oder b = (a|b)

- Mit Klammern und | ("oder") kann man gleichzeitig nach verschiedenen Wörtern suchen:

```
tok = /(Mann|Frau|Kind)/
```

Was ist mit *beßer*? Suche in der clean- oder norm-Ebene!

- Nach verschiedenen Formen:

```
tok = /(Mann|Mannes)/
```

- Oder Zeichenketten:

```
tok =/bes(ser|t).?/
```

besser, bessere
best, beste
Und andere Strings: *bests...*

33

Aufgabe 7

- **Finden Sie alle Formen des Nomens *Kraut* im Plural**
- **Welche Formen können in älteren Texten vorkommen?**
  - **Kräuter, kräuter, Kräutter, kräutter, Kreuter, kreuter, Kreutter, kreutter, Kråutern, Kreutern, kreüter, Kråuteren...**
  - **(ohne Komposita)**

34

Lösung 7

Nicht im aktuellen ridges v.1

```
tok=/(K|k)rr?(eu|äu|e͛u|å͛u|eü|aü)tt?e?r?e?n?.?/
```

oder

```
tok=/(K|k)rr?(e(u|ü)|(äu|aü)|å͛(u|ü))tt?e?r?e?n?.?/
```

→**Häufig gibt es alternative Suchanfragen für dieselben Treffermengen.**
→**Unbekannte Schreibweisen können nicht antizipiert werden! (*krühter)***

Einfachere Suche auf Norm oder Lemma → Export

35

Beispiel 8

- **Man möchte alle konjugierten Formen des Verbs *meinen* im Präsens finden, aber keine anderen Formen.**

mein	e
mein	st
mein	t
mein	en
mein	t
mein	en

tok =/mein(s?t|en?)/

oder

tok =/mein(e|st|t|en)/

➢**Problem Homograph:**
- **findet auch Infinitiv**
- **und Formen des Possessivpronomens**

36

Prinzip II: Relationen

- Einzelne **V**ariable-**W**ert-Paare werden durch "&" verbunden.
- Zwischen den Paaren muss **IMMER eine Beziehung hergestellt** werden
- Auf die VW-Paare bezieht man sich mit # der Reihe nach.

$Variable_1 = Wert_1$ & ← Ausdruck 1: #1
$Variable_2 = Wert_2$ & ← Ausdruck 2: #2
#1 "Beziehung" #2

37

Prinzip II: Relationstypen

$Variable_1 = Wert_1$ & ← Ausdruck 1: #1
$Variable_2 = Wert_2$ & ← Ausdruck 2: #2
#1 "Beziehung" #2

Operator	Description	Illustration	Notes
.	direct precedence	A B	For non-terminal nodes, precedence is determined by the right most and left most terminal children
.*	indirect precedence	A x y z B	For specific sizes of precedence spans, .n,m can be used, e.g. .3,4 - between 3 and 4 token distance
_=_	identical coverage	A B	Applies when two annotation cover the exact same span of tokens
_i_	inclusion	AAA B	Applies when one annotation covers a span identical to or larger than another
>	direct dominance	A | B	A specific edge type may be specifed, e.g.: >secedge to find secondary edges. Edges labels are specified in brackets, e.g. >[func="OA"] for an edge with the

Zu Beispiel 8

- Finden Sie nun Vorkommen von tok =/mein(e|st|t|en)/ , die ausschließlich finite Vollverben sind.

tok =/mein(e|st|t|en)/ & ← Ausdruck 1: #1
pos= "VVFIN" & ← Ausdruck 2: #2

#1_=_#2 ← Relation: Abdeckung

39

Negation !=

- **!** bedeutet Negation
  - Der Operator wird vor dem "="-Zeichen eingefügt.
  - **Finden Sie in alle Vorkommen von tok =/mein(e|st|t|en)/ , die nicht als finites Vollverb getaggt sind**

tok=/mein(e|st|t|en)/ &
pos!=/VVFIN/ &
#1 _=_ #2

40

Suche nach Abfolgen: z.B. Nomen folgt auf "zu"

tok = "zu"
&
pos = "NN"
&
#1 . #2

Ich	bin	zu	Hause
PPER	VAFIN	APPR	NN
Satz			
Subj		Adv	

41

Tokenfolgen – Aufgabe 9

- **Suchen Sie nach zwei aufeinanderfolgenden Adjektiven.**
- Achtung: Es gibt zwei Typen von Adjektiven
  - ADJA & ADJD

```
pos = /ADJ./ &
pos = /ADJ./ &
#1.#2
```

42

Metadaten

- Sie können Texte nach Metadaten filtern. Metadaten finden Sie unter dem i-Button in den Suchergebnissen
- Metadaten müssen nur verknüpft, aber nicht relatiert werden
- Beispiel: alle Token von Texten, die in Frankfurt publiziert wurden

```
tok &
meta::pubPlace="Frankfurt"
```

43

Tokenfolgen und Metadaten

- **Suchen Sie nach zwei aufeinanderfolgenden Adjektiven in einem Text von 1603!**

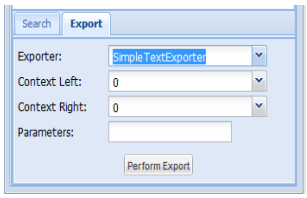
```
pos = /ADJ./ &
pos = /ADJ./ &
#1.#2 &
meta::date="1603"
```

44

Beispiel Export

- Sie suchen alle Schreibvarianten von *dass*
- Dazu können sie
  - nach lemma= "dass" suchen
  - die gefundenden Ergebnisse zur besseren Betrachtung aller Vorkommen exportieren

45



Ergebnisse (Auszug):

1. [daſ]	21. [daß]	80. [daſs]
2. [daſſ]	22. [dass]	81. [daſs]
3. [daſſ]	23. [daß]	82. [daſs]
4. [daſſ]	24. [daß]	83. [daſs]
5. [daſſ]	25. [daß]	84. [daſs]
6. [dz]	26. [daß]	85. [daß]
7. [daſ]	27. [dass]	319. [Das]
8. [daſ]	28. [dass]	320. [daſz]
9. [daſ]	29. [dass]	321. [daß]
10. [Das]	30. [daß]	322. [das]

46

Zusammenfassung Operatoren

- . Ein beliebiges Zeichen
- ? 0 oder 1 Zeichen (des vorherigen Elementes)
- * 0 bis unendlich viele Zeichen (d. vorh. E.)
- + 1 bis unendlich viele Zeichen (d. vorh. E.)
- \\ wörtlich (folgendes Zeichen)
- ! nicht
- [abc] Menge (oder [^abc]=alles außer abc)
- (a|b) a oder b (auch: [ab])
- a{2,3} a 2 bis 3mal

49

Zusammenfassung Relationen

- & verbindet Suchanfragen
- #1.#2 direkte Präzedenz
- #1.*#2 indirekte Präzedenz
- #13,5#2 #2 folgt mit 3 bis 5 Einheiten Abstand
- #1_=_#2 #1 und #2 deckungsgleich
- #1_o_#2 #1 und #2 überlappen sich
- #1_i_#2 #1 inkludiert #2
- Weitere Informationen
  - http://www.sfb632.uni-potsdam.de/d1/annis/

50

Vielen Dank!

51