FAU Erlangen-Nürnberg
Department Germanistik und Komparatistik
Lehrstuhl für Germanistische Sprachwissenschaft
Andreas Blombach

Stand: 28.02.2017

Anleitung zur Benutzung von Korpora zu geschriebenem und gesprochenem Deutsch

Inhaltsverzeichnis

1 Einleitung

Zum geschriebenen Deutsch gibt es eine Vielzahl von Korpora, die sich u. a. in ihrer Größe, ihrer Zielsetzung (und damit verbunden in der Textauswahl) und ihrer linguistischen Aufarbeitung deutlich voneinander unterscheiden. Zu den größten und bekanntesten Korpora gehören die Korpora des DWDS (Digitales Wörterbuch der deutschen Sprache) und die Korpora des IdS (Institut für deutsche Sprache).

Sowohl DWDS als auch IdS bieten ebenfalls Korpora der gesprochenen Sprache an, die jedoch deutlich kleiner ausfallen als die Korpora zur geschriebenen Sprache. Darüber hinaus sind viele der Einzelkorpora sehr speziell, was ihre Nutzbarkeit schmälert.

2 DWDS

Ein großer Teil der DWDS-Korpora ist unter http://dwds.de/r durchsuchbar. Ebenso findet man dort eine Übersicht über die einzelnen Korpora des Projekts (http://dwds.de/d/korpora). Leider sind selbst nach Registrierung nicht sämtliche Korpora online verfügbar, da einige aus rechtlichen Gründen nur für den projektinternen Gebrauch vorgesehen sind (darunter große Zeitungskorpora von BILD, WELT und SZ). Für die meisten Zwecke am interessantesten ist das Referenzkorpus des DWDS, das Kernkorpus des 20. Jahrhunderts (http://dwds.de/ressourcen/kernkorpus/). Dieses Korpus enthält über 79 000 Texte (ca. 100 Mio. Wortformen) aus den Jahren 1900 bis 1999 aus den Bereichen Gebrauchsliteratur, Belletristik, Wissenschaft und Zeitung; bei der Auswahl der Texte wurde auf eine ausgewogene Verteilung über die Jahrzehnte und Textsorten geachtet.[1] Ein Kernkorpus des 21. Jahrhunderts befindet sich gerade im Aufbau und ist derzeit noch nicht ausgewogen.

Ebenfalls durchsuchbar sind folgende Korpora:

- Deutsches Textarchiv (DTA): Im Gegensatz zu der Version auf der Hauptseite unter http://www.deutschestextarchiv.de (rund 2 500 Texte von 1600 bis 1900) enthält die Version auf dwds.de auch Texte, die sich im DTA noch in der Qualitätskontrolle befinden. Somit lassen sich über 3 200 Texte von 1465 bis 1927 durchsuchen (aktuell ca. 170 Mio. Wortformen). Zu beachten ist bei Untersuchungen stets die Korpuszusammensetzung: So stehen etwa 1 361 Texten aus dem 19. Jahrhundert 773 aus dem 17. und 107 aus dem 16. gegenüber (die Zahlen der Wortformen weichen noch deutlicher voneinander ab), und die einzelnen Textsorten sind in verschiedenen Jahrzehnten und Jahrhunderten sehr unterschiedlich stark vertreten (was natürlich z. T. in der Natur der Sache liegt).

- Drei große Zeitungskorpora: Die Zeit von 1946 bis heute (aktuell ca. 460 Mio. Wortformen), Berliner Zeitung von 1994 bis 2005 (ca. 200 Mio. Wortformen) und Tagesspiegel von 1996 bis 2005 (ca. 135 Mio. Wortformen).

- Das Korpus „Referenz- und Zeitungskorpora“ vereint die beiden Kernkorpora, das DTA sowie die drei Zeitungskorpora. Es enthält damit ca. 1,1 Milliarden Wortformen, allerdings zum Preis starker Unausgewogenheit.

[1] Die Daten des Kernkorpus verwendet auch das dlexDB-Projekt (http://www.dlexdb.de). Hiermit lassen sich z. B. Lemma- oder Silbenfrequenzen abfragen, ohne dass dabei aber die Einzeltreffer mit Kotext zurückgegeben werden. Interessant ist dies etwa für psycholinguistische Fragestellungen.

- Mehrere Spezialkorpora: Blog-Beiträge und Kommentare (aktuell ca. 90 Mio. Wortformen), Filmuntertitel (ca. 63 Mio. Wortformen), Polytechnisches Journal (ca. 68 Mio. Wortformen), DDR-Korpus (ca. 7 Mio. Wortformen), gesprochene Sprache. Das Korpus der gesprochenen Sprache besteht aus sieben Teilkorpora im Umfang von 200 000 bis 450 000 Wortformen; insgesamt erreicht die Korpussammlung eine Größe von über 2,25 Millionen Wortformen. Enthalten sind darin u. a. Transkripte von Reden (von Kaiser Wilhelm über Hitler bis Honecker) und alten Rundfunkansprachen (1929-1944), Auszüge aus Spiegel-Interviews und Parlamentsprotokollen sowie Auszüge aus der Fernsehsendung „Das Literarische Quartett“ (genauere Auflistung: http://dwds.de/d/k-spezial).

Eine ältere Version der Website ist unter http://eins.dwds.de erreichbar. Hier lassen sich z. B. Ergebnisse in mehreren Korpora gleichzeitig anzeigen.

2.1 Einfache Suche

Nach einzelnen Wörtern kann direkt gesucht werden, sie werden dabei jedoch automatisch expandiert – d. h., eine Suche nach `Kind` findet außer *Kind* auch *Kinder*, *Kindes* usw., also alle Realisierungsformen des Lexems.[2]

Will man dagegen nur nach einer ganz bestimmten Wortform suchen, muss man dieser ein `@` voranstellen: `@Kindern`.

Unter dem Eingabefeld für den Suchausdruck kann man das Korpus auswählen, in dem gesucht werden soll, den Suchzeitraum einstellen (z. B. nur Texte von 1990 bis 1999) und – bei den Referenzkorpora – Textklassen ein- oder ausschließen. Darüber hinaus lässt sich festlegen, wie die Ergebnisse sortiert und wie viele Treffer auf einmal angezeigt werden sollen.

Klickt man auf das Fragezeichen rechts oben (vgl. Abb. 1), ruft man die Suchhilfe auf.

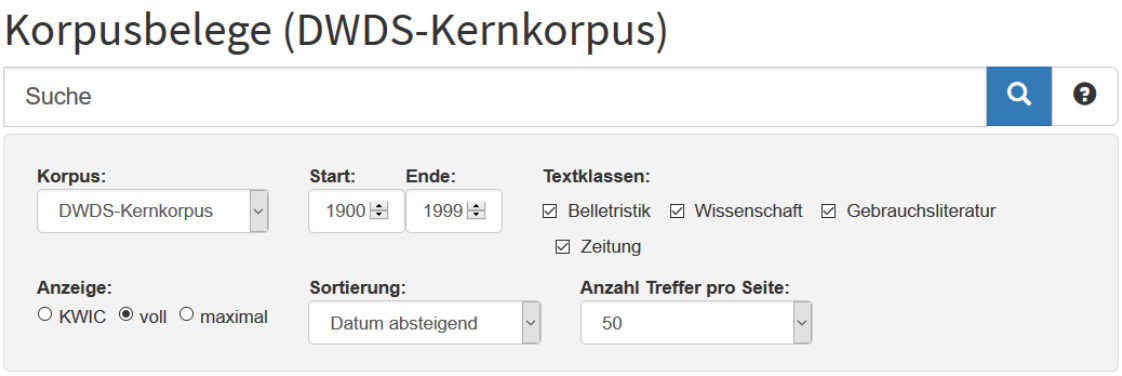


Abbildung 1: Sucheinstellungen im DWDS

Standardmäßig werden Ergebnisse im Satzkontext angezeigt („voll“), man kann sie sich jedoch auch in der kompakteren KWIC-Ansicht (KWIC steht für *key word in context*, gemeint ist damit, dass der Suchausdruck zusammen mit jeweils X Wörtern nach links und rechts angezeigt wird) oder mit mehr Kontext („maximal“) anzeigen lassen.

[2] In der vorigen DWDS-Version war es mitunter sinnvoller, sich nicht darauf zu verlassen, sondern mit `$l=` gezielt nach den konkreten Realisierungsformen eines Lexems zu suchen: `$l=Auftrag` (eine direkte Suche nach `Auftrag` fand z. B. *Aufträge* im Kernkorpus nicht). Mit `$l=` gibt man dabei die Zitierform des Lexems an, also das Lemma.

2.2 Ergebnisse speichern

Über und unter den Ergebnissen sieht man nicht nur die Trefferzahl, sondern auch die Exportmöglichkeit: „[Export als: TSV]“. Klickt man auf „TSV“, wird eine Textdatei erzeugt, die die Treffer der aktuellen Seite mitsamt Publikationsdatum, Textklasse und bibliographischen Angaben enthält. Verändert man die Adresszeile im Browser, lassen sich auch mehr als 100 Belege auf einmal speichern, nämlich bis zu 5 000: Dazu ändert man einfach den Wert bei „limit=“.

2.3 Suche mit Platzhalter

Will man nach Wörtern suchen, die eine bestimmte Zeichenkette enthalten, kann man einen sog. Platzhalter (engl. *wildcard*) verwenden: Mit einem Asterisk kann man angeben, dass an dieser Stelle beliebig viele Zeichen stehen können (auch 0!). So liefert eine Suche nach `*acht*` u. a. Treffer für *acht*, *belacht* und *beobachtet*.

2.4 Suche nach Wortfolgen

Nach einfachen Wortfolgen kann man suchen, indem man sie in Anführungszeichen setzt: `"eine Wortfolge"`. Dabei ist zu beachten, dass alle Einzelwörter automatisch expandiert werden; man muss also ggf. `@` verwenden. Über Satzgrenzen hinweg kann man nicht suchen.

Außerdem lassen sich Wortplatzhalter verwenden, nämlich `#2` (maximal zwei Wörter), `#>2` (mindestens zwei Wörter) und `#=2` (genau zwei Wörter).

Wenn die Reihenfolge der gesuchten Wörter egal ist, kann man mit `near()` suchen, z. B. nach den Wörtern *Shitstorm* und *Internet* mit maximal vier Wörtern dazwischen: `near(Shitstorm, Internet, 4)`.

2.5 Suche nach Wortarten

Alle DWDS-Korpora sind nicht nur lemmatisiert, sondern auch wortartenannotiert (nach dem STTS-Tagset: http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html). Das ermöglicht es, gezielt nach Wörtern zu suchen, die zu einer bestimmten Wortart gehören. Sucht man z. B. nach einer Wortfolge aus einem beliebigen attributiven Adjektiv, gefolgt von einer Flexionsform von *Erfolg*, kann man die Suchanfrage folgendermaßen formulieren: `"$p=ADJA $l=Erfolg"`.

Mit dem Operator `with` (auch: `&=`) kann man die Wortart eines gesuchten Wortes spezifizieren. Beispiel: *der* als substituierendes Relativpronomen: `@der with $p=PRELS`.

Das Gegenteil zu `with` ist `without` (auch: `!with`, `!=`, `&!=`). Damit lässt sich festlegen, dass ein Wort nicht einer bestimmten Wortart angehören soll, z. B. das Wort *schöneres*, aber nicht als attributives Adjektiv: `@schöneres without $p=ADJA`.

Bei den Wortarten-Tags lässt sich der Platzhalter `*` verwenden: `$p=ADJ*` findet z. B. attributiv, adverbial und prädikativ gebrauchte Adjektive (ADJA und ADJD). Leider funktioniert dies nur am Ende des Tags, nicht im Inneren. Dafür ist jedoch ein regulärer Ausdruck möglich – siehe Abschnitt 2.8.

2.6 Position im Satz

Setzt man hinter ein Suchwort `with $.=`, gefolgt von einer positiven oder negativen Zahl, gibt man damit die Position im Satz an, an der das Suchwort stehen soll. Mit positiven Zahlen zählt man Wörter (und Satzzeichen) vom Satzanfang an (0 für das erste Wort[3], 1 für das zweite, 2 für das dritte usw.), mit negativen Zahlen zählt man vom Satzende (-1 für das Satzzeichen am Satzende[4], -2 für das letzte Wort vor dem Satzzeichen, -3 für das vorletzte usw.). Beispiele:

- *Minister* an dritter Position im Satz: `@Minister with $.=2`
- Fragezeichen am Satzende: `\? with $.=-1`[5]
- ein finites Voll-, Hilfs- oder Modalverb am Satzende:
  `$p=VVFIN with $.=-2 || $p=VAFIN with $.=-2 || $p=VMFIN with $.=-2`

Um Positionen auszuschließen, lässt sich auch hier `without` verwenden. Der Ausdruck `\? without $.=-1 without $.=-2` findet z. B. Sätze, in denen irgendwo ein Fragezeichen vorkommt, allerdings nicht am Ende (oder an vorletzter Position, etwa vor schließenden Anführungszeichen).

2.7 Logische Verknüpfungen

Drei logische Operatoren lassen sich verwenden, nämlich `&&` (und), `||` (oder) und `!` (nicht). So kann man z. B. nach allen Sätzen im Korpus suchen, die die Wörter *Twitter* und *Shitstorm*, nicht aber *Unternehmen* enthalten: `Twitter && Shitstorm && ! Unternehmen`. Rücksicht auf die Reihenfolge oder den Abstand der Wörter im Satz wird dabei nicht genommen.

Für Wortalternativen innerhalb von Wortfolgen bietet sich – neben regulären Ausdrücken – der Operator `withor` an (auch: `wor`, `|=`):
`"kein Grund zur Panik withor Sorge withor Beunruhigung"` sucht nach Sätzen, in denen auf *kein Grund zur* entweder *Panik* oder *Sorge* oder *Beunruhigung* folgt. Alternativ kann man die Alternativen auch in geschweifte Klammern setzen und mit Kommata trennen: `"kein Grund zur {Panik, Sorge, Beunruhigung}"`.

2.8 Reguläre Ausdrücke

Besonders mächtig ist die Suche mit regulären Ausdrücken, die mit Schrägstrichen umschlossen werden: `/Groß/` (Alternativschreibungen mit *ss* werden damit nicht mehr gefunden, mit `/Gro(ß|ss)/` dagegen schon). Zu beachten ist dabei, dass standardmäßig alle Wörter gefunden werden, die den angegebenen Ausdruck *enthalten*, sodass dieser also bspw. auch *Großbritannien* findet. Will man das nicht, kann man entweder die Zeichen `^` und `$` für Wortanfang bzw. Wortende verwenden (s. u.) oder `g` nach dem schließenden Schrägstrich schreiben, um nur Wörter zu finden, die dem Ausdruck exakt entsprechen: `/Groß/g`. Analog

[3] Auch wenn 0 strenggenommen natürlich keine positive Zahl ist …

[4] Einige wenige Wörter kommen auch hier vor, z. B. in Überschriften.

[5] Dem Fragezeichen wird hier ein Backslash vorangestellt, weil es sonst als Suchoperator interpretiert wird. Der Backslash ist ein sog. Maskierungs- oder Fluchtzeichen (*escape character*) und lässt sich in allen hier vorgestellten Korpussammlungen verwenden, um Zeichen mit Spezialfunktionen entsprechend zu „maskieren“.

lässt sich mit `i` signalisieren, dass Groß- und Kleinschreibung egal sein sollen: `/Groß/i` findet also auch *groß*. Kombinieren lassen sich solche Modifikatoren ebenfalls: `/Groß/gi`. Stellt man dem öffnenden Schrägstrich ein Ausrufezeichnen voran, wird der reguläre Ausdruck negiert, findet also nur noch Wörter, die ihn *nicht* enthalten: `!/^ver/ with $p=VVFIN` findet finite Verben, die nicht mit *ver-* beginnen. Allerdings funktioniert dies derzeit nicht in jedem Korpus – insbesondere dem Zeit-Korpus – zuverlässig. Oft kann man sich aber mit `without` behelfen: `$p=VVFIN without /^ver/` bewirkt das gleiche.

Reguläre Ausdrücke sind auch bei der Angabe der Wortart erlaubt, was z. B. gestattet, die Suche nach finiten Verben am Satzende (siehe Abschnitt 2.6) etwas kürzer zu gestalten: `$p=/V.FIN/ with $.=-2`.

Es ist nicht Ziel dieser Ausführungen, reguläre Ausdrücke zu erklären oder eine umfangreiche Einführung zu bieten (dazu gibt es zahlreiche Tutorials im Internet), daher soll die folgende kurze (und unvollständige) Aufzählung von Möglichkeiten genügen:

- Ein Punkt steht für ein beliebiges Zeichen. Sucht man nach einem Punkt, muss man diesem einen Backslash voranstellen: `\.`. Das gilt genauso für sonstige Zeichen mit Sonderfunktionen.

- Setzt man eckige Klammern um eine Menge von Zeichen, so heißt dies für den regulären Ausdruck, dass an dieser Stelle irgendeines dieser Zeichen stehen soll. Die Auswahl `[aeiouüäö]` z. B. passt auf irgendeines der Vokalgrapheme in der Auswahl (allerdings nicht auf großgeschriebene – dazu bräuchte man `[aeiouüäöAEIOUÜÄÖ]`). Mit einem Bindestrich lassen sich Zeichenbereiche definieren: `[a-z]` erfasst bspw. alle Kleinbuchstaben von *a* bis *z* (ohne Umlaute!), `[a-zA-Z0-9äöüÄÖÜß]` erfasst Klein- und Großbuchstaben, Ziffern und *ß*. Alternativ lassen sich auch vordefinierte Zeichenklassen nutzen – so steht etwa `[[:alpha:]]` für Klein- und Großbuchstaben, `[[:lower:]]` für Klein-, `[[:upper:]]` für Großbuchstaben und `[[:punct:]]` für Satz- und Sonderzeichen.

  Stellt man der Zeichenmenge in eckigen Klammern einen Zirkumflex voran, negiert man sie damit – erfasst werden dann alle Zeichen, die *nicht* in der Menge enthalten sind: `[^a-z]` erfasst alle Zeichen außer *a* bis *z*.

- Quantoren geben an, wie oft der vorangehende Ausdruck vorkommen soll:
  - `?`: einmal oder keinmal (`Hunde?` findet *Hund* und *Hunde*)
  - `+`: mindestens einmal bis unendlich oft (`nei+n` findet *nein*, *neiin*, *neiiin* usw.)
  - `*`: beliebig oft (auch keinmal)
  - `{n,m}`: mindestens n-mal, maximal m-mal (`nei{2,4}n` findet *neiin*, *neiiin* und *neiiiin*; wird die Obergrenze nach dem Komma weggelassen, muss der Ausdruck mindestens n-mal bis unendlich oft vorkommen, wird auch noch das Komma weggelassen, muss der Ausdruck exakt n-mal vorkommen)

  Der vorangehende Ausdruck kann dabei ein einzelnes Zeichen, eine Zeichenmenge in eckigen Klammern oder ein womöglich komplexerer Ausdruck in runden Klammern sein (letztere haben nur die Funktion, Ausdrücke zusammenzufassen): `[a-z]+` findet beliebig lange Kombinationen aus den Buchstaben *a* bis *z*, `(eine?)?` findet *ein* und *eine*, es darf aber auch gar nichts an dieser Stelle stehen, da der ganze Ausdruck umklammert und mit `?` markiert wurde.

- Alternative Ausdrücke werden mit `|` eingeleitet: `Mann|Frau` passt auf *Mann* und *Frau*, `(ver|be)enden` auf *verenden* und *beenden*.

- `^` und `$` stehen normalerweise für Zeilenanfang bzw. -ende, im DWDS lassen sich damit jedoch Wortanfang bzw. -ende markieren. Die Abfrage `/^[[:alpha:]]{2}$/` findet im DWDS z. B. alle Wörter, die aus genau zwei Buchstaben bestehen.

Eine komplexere Abfrage mit regulären Ausdrücken kann im DWDS z. B. so aussehen: `"@Was $p=VVFIN /.+/ #7 @für /(eine?)?/"`. Gefunden werden damit Wortfolgen, die mit der konkreten Wortform *Was* beginnen, gefolgt von einem finiten Vollverb, einem beliebigen Wort und der konkreten Wortform *für*, wobei dazwischen bis zu sieben weitere Wörter stehen dürfen, sowie – optional – der Wortform *ein* oder *eine*. Unter den Treffern befinden sich dann Sätze wie *Was stellte er denn für Fragen!*, *Was spielt es für eine Rolle?*, aber auch *Was bedeutet es für Sie, im Leistungsvergleich mit Kollegen gemessen zu werden?* (was womöglich nicht der gesuchten Konstruktion entspricht). Genau wird sich die *Was-für*-Konstruktion mit einer DWDS-Abfrage nicht erfassen lassen, aber Verbesserungen sind zweifellos möglich. So kann man etwa fordern, dass nach *für* (und ggf. *ein* oder *eine*) in einem gewissen Abstand noch ein Substantiv folgt, und man kann ausschließen, dass Verben wie *bedeuten* im Satz vorkommen: `"@Was $p=VVFIN /.+/ #7 @für /(^eine?$)?/ #1 $p=NN" && ! $l=bedeuten`.

2.9 Wortverlauf

Häufig ist es interessant, die Häufigkeit eines Wortes (oder eines komplexeren Ausdrucks) im zeitlichen Verlauf zu betrachten. Unter http://www.deutschestextarchiv.de/search/plot lassen sich dazu die Ergebnisse aus dem DWDS-Kernkorpus des 20. Jahrhunderts und dem Deutschen Textarchiv zusammengefasst visualisieren.[6]

Abb. 2 zeigt beispielhaft den Wortverlauf von `Frieden`, getrennt nach Textklassen. Dabei ist allerdings zu beachten, dass die einzelnen Textklassen im Deutschen Textarchiv über die Jahrzehnte nicht gut ausbalanciert sind, dass sie also je nach Jahrzehnt stärker oder schwächer vertreten sind (so kommt es etwa dazu, dass für einige Jahrzehnte gar keine Zeitungstreffer angezeigt werden – hier liegen einfach keine Texte vor). Das verzerrt natürlich die Darstellung und macht den Bereich vor 1900 unzuverlässiger als den danach.

Unter http://dwds.de/stats lässt sich auf Basis der Zeitungs- und Referenzkorpora eine Wortverlaufskurve erstellen, die die Zeit vom 17. Jahrhundert bis heute berücksichtigt. Aufgrund der Unausgewogenheit der Korpusbasis ist Vorsicht bei der Interpretation der Ergebnisse geboten.

Wortverläufe in den einzelnen DWDS-Korpora lassen sich ebenfalls erstellen. Man muss hierzu allerdings auf die alternative Eingabemaske zurückgreifen, die in Abschnitt 2.11 kurz beschrieben wird.

Bei all diesen Wortverläufen beeinflussen die gewählten Einstellungen extrem, wie die Kurven letztendlich aussehen – es ist nicht immer sinnvoll, sich auf die Standardeinstellungen zu verlassen! Um zu verstehen, was welche Option bewirkt, lohnt sich die Auseinandersetzung mit der Kurzdokumentation unter http://dwds.de/d/plot (leider aktuell noch sehr technisch).

[6] Ganz ähnlich wie beim „Ngram Viewer“ von Google Books: https://books.google.com/ngrams/graph?content=Frieden&year_start=1800&year_end=2008&corpus=20&smoothing=2.

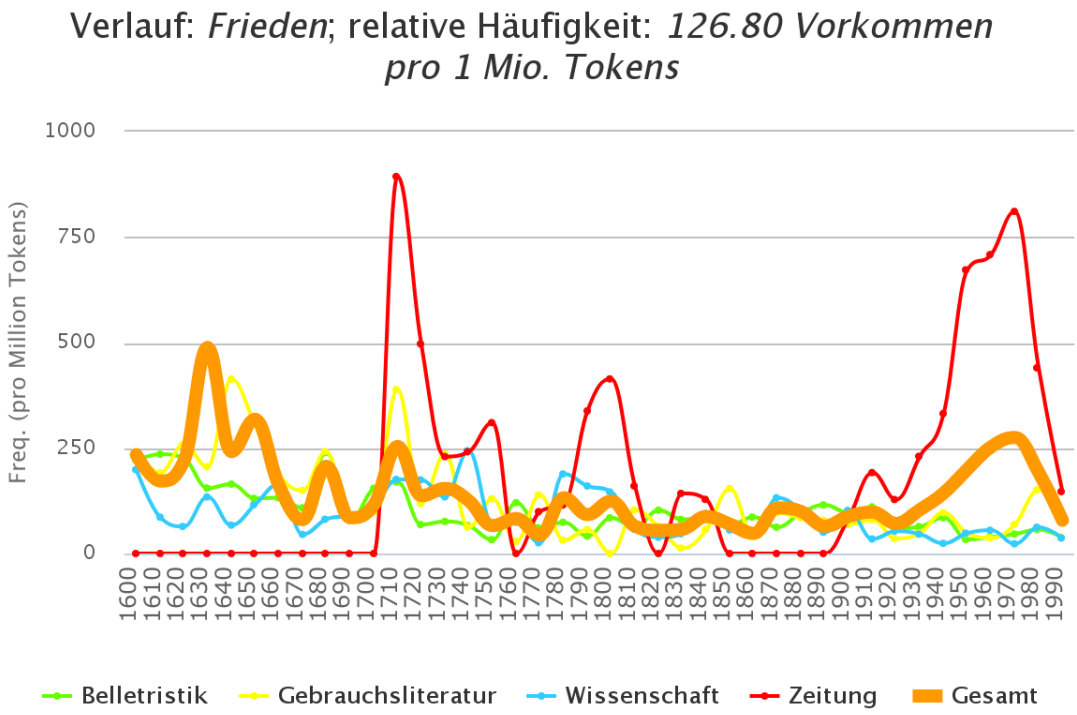


Abbildung 2: Wortverlauf von `Frieden`

2.10 Kookkurrenzanalyse

Kookkurrenz- oder Kollokationsanalysen kann man durchführen, wenn man wissen möchte, welche Wörter besonders häufig im Umfeld des Suchbegriffs vorkommen. Zur Ermittlung besonders auffälliger Kookkurrenzen, also gemeinsamer Vorkommen, gibt es verschiedene Assoziationsmaße, die die Stärke der Verbindung angeben.[7] Diese verschiedenen Verfahren liefern z. T. sehr unterschiedliche Ergebnisse.

Unter http://dwds.de/stats lassen sich das DWDS-Wortprofil und DiaCollo aufrufen. Das Wortprofil liefert Kollokationskandidaten[8] zu einem Suchwort (auch unterteilt nach grammatischen Relationen wie Genitiv- oder Adjektivattribut), wobei es auch möglich ist, zwei Wörter miteinander zu vergleichen, um Gemeinsamkeiten und Unterschiede bei den häufigen Kookkurrenzpartnern festzustellen. Mit DiaCollo (Kollokationsanalyse in diachroner Perspektive) kann man dagegen Kollokationskandidaten zu einem Suchwort im Wandel der Zeit betrachten. Eine Kurzanleitung sowie ausführliche Informationen gibt es unter http://clarin-d.de/de/kollokationsanalyse-in-diachroner-perspektive.

2.11 Frequenzlisten

Will man bspw. verschiedene Treffer nach ihrer Frequenz im Korpus sortieren, so ist dies unter dwds.de/r derzeit leider noch nicht möglich. Es gibt allerdings eine alternative Eingabemaske für Abfragen, zu erreichen unter:

- http://kaskade.dwds.de/dstar/kern (Kernkorpus)

[7] Siehe hierzu bspw. http://www.collocations.de/AM/index.html oder https://nlp.fi.muni.cz/raslan/2008/papers/13.pdf.

[8] Hierfür wird das Assoziationsmaß logDice verwendet.

- http://kaskade.dwds.de/dstar/korpus21 (Kernkorpus 21)
- http://kaskade.dwds.de/dstar/dta (Deutsches Textarchiv)
- http://kaskade.dwds.de/dstar/bz (Berliner Zeitung)
- http://kaskade.dwds.de/dstar/tagesspiegel (Tagesspiegel)
- http://kaskade.dwds.de/dstar/zeit (Die Zeit)
- http://kaskade.dwds.de/dstar/blogs (Blogs)
- http://kaskade.dwds.de/dstar/dingler (Polytechnisches Journal)
- http://kaskade.dwds.de/dstar/untertitel (Filmuntertitel)

Hier sind dieselben Abfragen möglich, nur dass man ihnen grundsätzlich `#sep` nachstellen sollte (z. B. `Panik #sep`), damit mehrere Treffer innerhalb desselben Satzes separat aufgeführt werden.

Darüber hinaus lässt sich nach einer durchgeführten Abfrage mit einem Klick auf „~Hist“ ein Wortverlauf erzeugen (was in jedem Korpus funktioniert).

Für Frequenzen gibt es den Befehl `count()`. Mit den Klammern umschließt man einfach eine Abfrage: `count(Panik #sep)` (mit `count(* #sep)` zählt man alle Tokens inklusive Satzzeichen im aktuellen Korpus, mit `count(*)` alle Sätze).

Richtig interessant wird dies für Abfragen, die verschiedene Trefferwörter zurückliefern. Mit der Ergänzung `#by[]` lässt sich angeben, wonach beim Zählen gruppiert werden soll: `#by[date]` gruppiert nach dem Erscheinungsjahr, `#by[date/10]` nach dem Jahrzehnt, `#by[date/100]` nach dem Jahrhundert (Beispiel: `count(* #sep) #by[date/10]` gibt aus, wie viele Tokens es pro Jahrzehnt im Korpus gibt), `#by[$Token]` gruppiert nach Tokens und `#by[$l]` nach Lemmata.

Sehr häufig ist es sinnvoll, solche Listen nach Häufigkeit zu sortieren, was mit `#desc_count` (absteigend, alternativ: `#greater_by_count`) oder `#asc_count` (aufsteigend; alternativ: `#less_by_count`) geht.

Ganze Wortgruppen kann man m. W. aktuell zwar noch nicht nach ihrer Häufigkeit sortieren lassen, man kann jedoch die Häufigkeiten von Wörtern an einer bestimmten Position der Wortgruppe sortieren. `#by[$Token]` (oder `#by[$l]`) sortiert die Wörter, die an erster Stelle stehen, nach ihrer Häufigkeit, `#by[$Token +1]` die Wörter, die an zweiter Stelle stehen, `#by[$Token +2]` die Wörter, die an dritter Stelle stehen, usw.

Will man (z. B. im DTA) die Wörter nach Häufigkeit gruppieren, aber nur einen bestimmten Zeitraum betrachten (z. B. nur das 17. Jahrhundert), kann man der ursprünglichen Abfrage so etwas hinzufügen: `#asc_date[1600,1699]`.

Beispiele:

- Alle Verben absteigend nach Häufigkeit: `count($p=V*) #by[$l] #desc_count`
- Wörter auf *-ität*: `count(/^[[:alpha:]]+ität$/ #sep) #by[$Token] #desc_count` (angefangen bei den häufigsten)
- Wörter auf *-ität*: `count(/^[[:alpha:]]+ität$/ #sep) #by[$Token] #asc_count` (angefangen bei den seltensten)

- `count("kein Grund {zur, zum} $p=NN" #sep) #by[$Token +3] #desc_count` sortiert die Substantive, die nach *kein Grund zur* oder *kein Grund zum* kommen, absteigend nach ihrer Häufigkeit.
- `count(/^[[:alpha:]]+ism(us|en)$/ #sep #asc_date[1600,1699]) #by[$l] #desc_count` sucht nach Wörtern auf *-ismus* oder *-ismen*, die in Texten aus dem 17. Jahrhundert vorkommen, und sortiert die Lemmata absteigend nach ihrer Häufigkeit.

3 DeReKo/COSMAS II

Die IdS-Korpora, auch bekannt als DeReKo (Deutsches Referenzkorpus), bilden die derzeit größte Sammlung von Korpora zum Deutschen – insgesamt enthalten sie über 28 Milliarden Wortformen in über 340 Korpora, aufgeteilt in 17 Archive. Sie lassen sich – nach kostenloser Registrierung – mit dem COSMAS-System durchsuchen, insbesondere mit der Web-Anwendung COSMAS II_{web} (http://www.ids-mannheim.de/cosmas2/web-app/).

Eine Archivübersicht bieten http://www1.ids-mannheim.de/kl/projekte/korpora/archiv.html und http://www.ids-mannheim.de/cosmas2/projekt/referenz/korpora.html. Unter den enthaltenen Korpora befinden sich historische Korpora (u. a. historische Zeitschriften und Texte von Marx und Engels, Goethe und den Brüdern Grimm), diverse Regionalzeitungen, die sehr viele Regionen Deutschlands, Österreichs und der Schweiz abdecken, Zeitschriften (z. B. Focus) und überregionale Zeitungen (ZEIT, taz, SZ), Wikipedia-Artikel und -Diskussionen, das Wendekorpus (Sprache in West- und Ostdeutschland 1989/90), literarische Texte (u. a. von Christa Wolf und Martin Walser) und Sonderkorpora (etwa zu Fachsprachen). Den weitaus größten Teil der Korpora machen Zeitungskorpora aus, das Gesamtarchiv ist also keineswegs balanciert.

Leider ist nur ein kleiner Teil der Korpora auch wortartenannotiert – diese Korpora befinden sich in den Archiven TAGGED-T, TAGGED-T2 (beide mit dem STTS-Tagset annotiert), TAGGED-C, TAGGED-C2 und TAGGED-M. Groß sind diese Archive allerdings immer noch: das neue Archiv TAGGED-T2 mit Zeitungstexten von 2010 bis 2014 etwa enthält ca. 1,38 Milliarden Wortformen.

3.1 Korpusauswahl und virtuelle Korpora

Ehe man mit einer Suche beginnen kann, muss ein Archiv ausgewählt werden. Anschließend lädt man entweder alle Korpora dieses Archivs zusammen, wählt ein Einzelkorpus aus oder erstellt unter „geladene Korpora“ durch Klick auf „Neu“ ein benutzerdefiniertes, „virtuelles“ Korpus, indem man beliebige Einzelkorpora aus dem gewählten Archiv zusammenfügt (siehe Abb. 3). Wenn man an einem ganz bestimmten Sprachausschnitt interessiert ist, ist letzteres die beste Vorgehensweise.

3.2 Einfache Suche

Werden Wörter direkt eingegeben, so wird nach konkreten Wortformen gesucht (allerdings in allen Varianten von Groß- und Kleinschreibung) – sie werden also nicht automatisch expandiert. Eine Suche nach `Hund` findet nur Treffer für *Hund*, sonst nichts. Will man auch Flexionsformen erfassen, so kann man den Grundformoperator `&` verwenden: `&Hund` findet *Hund*, *Hunde*, *Hundes*, *Hunds*, *Hunden* und (sonderbarerweise; je nach Archiv) *Hundsche*

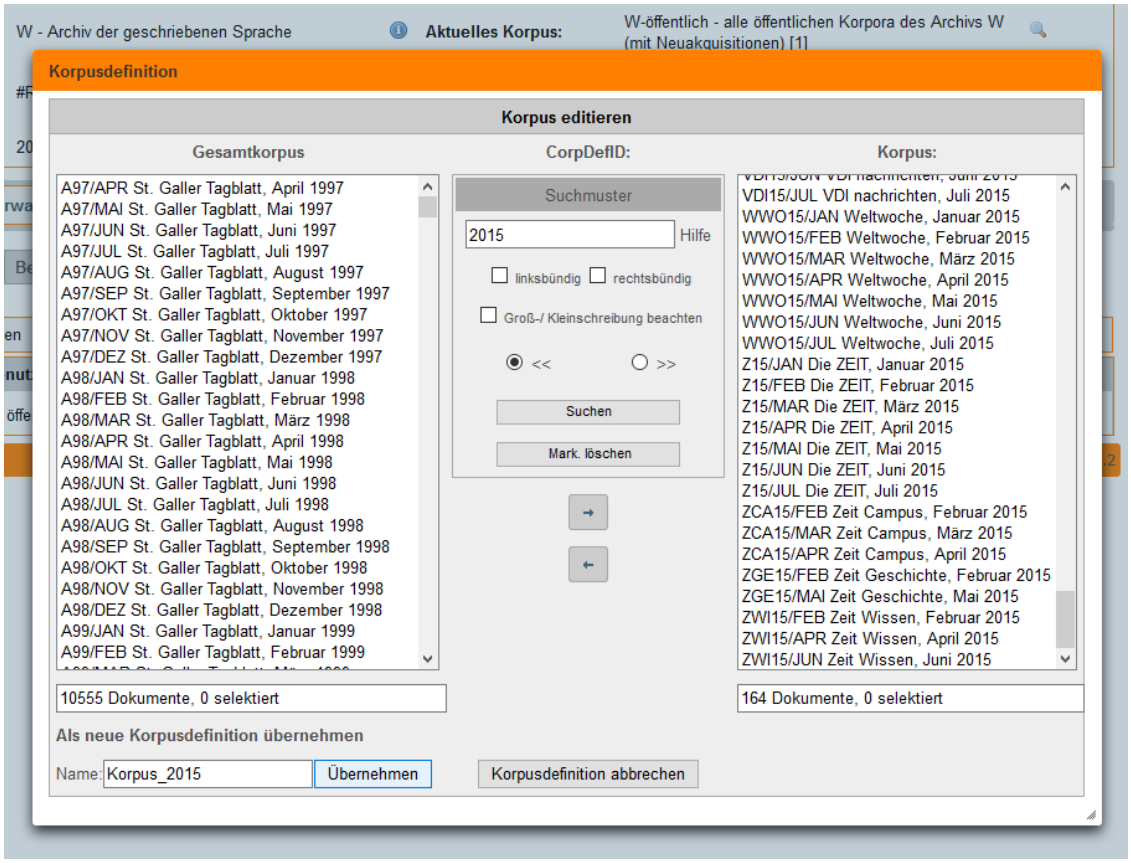


Abbildung 3: Erstellung eines virtuellen Korpus (hier mit Texten aus dem Jahr 2015)

sowie *Hundschen*. Allerdings wird nie direkt gesucht, stattdessen werden zunächst alle Wortformen präsentiert, die von der Suche erfasst werden. Hier kann man auch Wortformen abwählen und so von der Suche ausschließen, ehe man auf „Ergebnisse“ klickt.

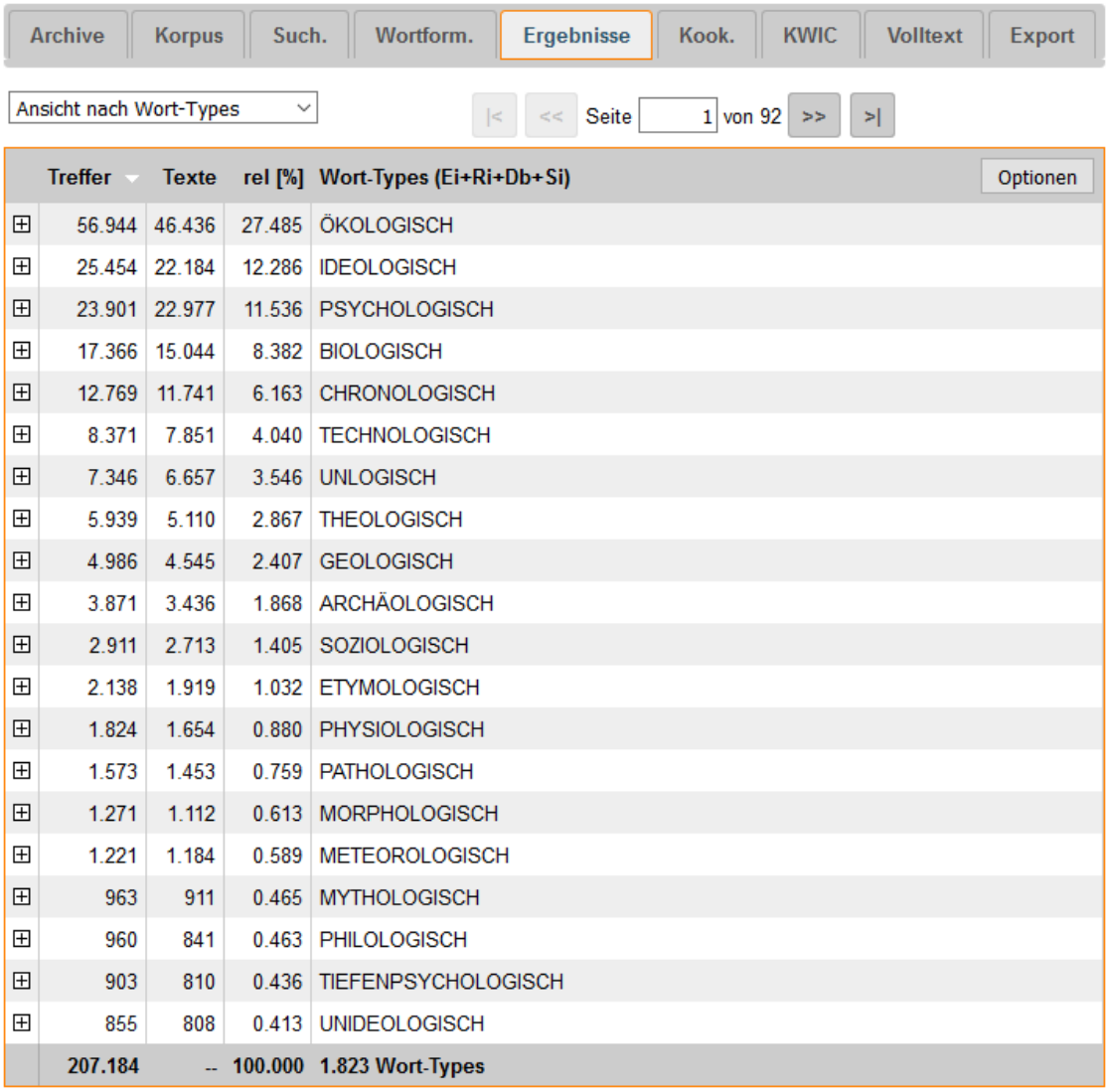
3.3 Ergebnisanzeige

Bei den Ergebnissen werden die Treffer nicht direkt angezeigt, sondern zunächst nur, wie viele Treffer in welchen Korpora erzielt wurden. Durch Klick auf „+“ werden dann konkrete Treffer angezeigt, standardmäßig in der KWIC-Ansicht (durch Klick auf „Volltext“ rechts oben lässt sich erheblich mehr Kontext anzeigen). Die Ergebnisse lassen sich auch anders gruppieren, z. B. nach Ländern, Textsorten, Themen oder Zeit (Jahrzehnt, Jahr, Monat, . . .). Dazu wählt man einfach über den Treffern etwas anderes als „Korpusansicht“ aus.

Hat man mit einer Suchanfrage nach mehreren verschiedenen Wörtern oder Ausdrücken gesucht, ist die „Ansicht nach Wort-Types“ besonders interessant. Hiermit lassen sich die verschiedenen Treffer nach ihrer Frequenz sortieren. Abb. 4 zeigt exemplarisch die Ergebnisse einer Suche nach Wörtern, die auf *-logisch* enden.[9] Bei den Optionen (oben rechts im Bild) wurde dafür nur „Diakritische Zeichen beachten“ gewählt, Groß- und Kleinschreibung sollte absichtlich nicht beachtet werden (deshalb sind alle Wörter in Großschreibung aufgeführt; andernfalls würden z. B. *Ökologisch* und *ökologisch* getrennt genannt). Durch einen Klick auf „Treffer“ wird nach absoluter Häufigkeit sortiert – zunächst aufsteigend, ein zweiter Klick

[9] Gesucht wurde mit dem regulären Ausdruck `#REG(^[[:alpha:]]+logisch$)` – siehe Abschnitt 3.11.

sortiert die Ergebnisse dann absteigend vom häufigsten Ergebnis zum seltensten.



Treffer	Texte	rel [%]	Wort-Types (Ei+Ri+Db+Si)
56.944	46.436	27.485	ÖKOLOGISCH
25.454	22.184	12.286	IDEOLOGISCH
23.901	22.977	11.536	PSYCHOLOGISCH
17.366	15.044	8.382	BIOLOGISCH
12.769	11.741	6.163	CHRONOLOGISCH
8.371	7.851	4.040	TECHNOLOGISCH
7.346	6.657	3.546	UNLOGISCH
5.939	5.110	2.867	THEOLOGISCH
4.986	4.545	2.407	GEOLOGISCH
3.871	3.436	1.868	ARCHÄOLOGISCH
2.911	2.713	1.405	SOZIOLOGISCH
2.138	1.919	1.032	ETYMOLOGISCH
1.824	1.654	0.880	PHYSIOLOGISCH
1.573	1.453	0.759	PATHOLOGISCH
1.271	1.112	0.613	MORPHOLOGISCH
1.221	1.184	0.589	METEOROLOGISCH
963	911	0.465	MYTHOLOGISCH
960	841	0.463	PHILOLOGISCH
903	810	0.436	TIEFENPSYCHOLOGISCH
855	808	0.413	UNIDEOLOGISCH
207.184	--	**100.000**	**1.823 Wort-Types**

Abbildung 4: Suche im Archiv W nach Formen auf *-logisch*

3.4 Ergebnisse speichern

Durch Klick auf „Export“ gelangt man zu einem Menü, in dem man auswählen kann, in welcher Form, mit wie viel Kontext und mit welchen zusätzlichen Informationen die Treffer gespeichert werden sollen.

3.5 Optionen

COSMAS II bietet viele Optionen zur Suche, Lemmatisierung und Präsentation, die sich sitzungsübergreifend speichern lassen und z. T. recht große Auswirkungen haben.

U. a. lässt sich hier einstellen, ob Groß- und Kleinschreibung bei der Suche beachtet werden soll, ob alle Ergebnisse ausgegeben werden sollen oder eine Zufallsausfall daraus, ob/welche

Häufigkeitsmaße für Ergebnisse berechnet werden sollen und wie weit der Kontext in der KWIC-Anzeige sein soll.

Besonders wichtig sind auch die Einstellungen zum Verhalten des Grundformoperators `&`: Außer Flexionsformen lassen sich damit nach Wahl auch Komposita, sonstige Wortbildungsformen (v. a. Derivationen) und Spezialfälle erfassen. Um das Beispiel der Suche nach `&Hund` fortzuführen: Erfasst der Grundformoperator auch Komposita, wird (im Korpus W-Öffentlich) nach über 15 000 Wortformen gesucht, darunter *hundähnlich*, *Landeshundehalterverordnung* oder *Unterwasserblindenhund*. Erfasst der Operator sonstige Wortbildungsformen, wird u. a. nach *abhunden*, *Biohund* und *hundlich* gesucht.[10] Spezialfälle schließlich sind v. a. Bindestrichformen wie *Mensch-Hund-Beziehungen*.

3.6 Suche mit Platzhaltern

Eine Reihe von Platzhaltern lässt sich bei der Suche verwenden:

- `*` steht wie im DWDS für beliebig viele Zeichen. `*lich` findet *herrlich*, *kindlich*, *möglich* usw.
- `+` steht für 0 oder 1 Zeichen. `Hund+` findet u. a. *Hund*, *Hunde* und *Hunds*, `++turm` findet u. a. *Turm*, *Sturm* und *Getürm* (weil COSMAS II nicht zwischen *u* und *ü* unterscheidet, wenn in den Optionen kein Haken bei „Diakritische Zeichen beachten“ gesetzt wurde).
- `?` steht für genau ein Zeichen.

Soll nach den Zeichen selbst gesucht werden, muss ihnen ein Backslash vorangestellt werden: `\?`.

3.7 Suche nach Wortarten

In den wortartenannotierten Korpora (wie TAGGED-T2) kommt bei der Suchanfrage ein neuer Button hinzu, „MORPH-Assistent“. Klickt man darauf, kann man bequem aussuchen, nach welcher Wortart man suchen möchte; die Auswahl wird nach Klick auf „Übernehmen“ ins Suchfenster eingefügt: `MORPH(ADJ at)` sucht dann bspw. nach attributiven Adjektiven. Die genauen Suchmöglichkeiten unterscheiden sich dabei etwas, je nachdem, mit welchem Tagset das Korpus annotiert wurde.

Soll ein bestimmtes Wort einer bestimmten Wortart angehören, kann man `/w0` (eigentlich ein Abstandsoperator) verwenden, um Wort und Wortart in der Abfrage zu verknüpfen. Wenn man z. B. *der* als substituierendes Relativpronomen sucht, kann man entsprechende Vorkommen im Korpus mit `der /w0 MORPH(PRON rel sub)` finden.

Seit Ende 2015 ist es möglich, dem MORPH-Operator einen Wiederholungsfaktor hinzuzufügen, mit dem es z. B. möglich ist, nach drei Adjektiven hintereinander zu suchen: `MORPH(ADJ){3}`. Der Ausdruck in geschweiften Klammern funktioniert dabei wie bei regulären Ausdrücken (siehe Abschnitt 2.8), also nach dem Muster {n,m} – mindestens n-mal, maximal m-mal.

[10] Damit kann auch nach Affixen gesucht werden, z. B. `&-heit` oder `&ver-`.

3.8 Suche mit Abstandsoperatoren

Um nach mehreren Wörtern zu suchen, verwendet man Abstandsoperatoren:

- Der Wortabstandsoperator `w` gibt an, wie viele Wörter maximal zwischen zwei Suchbegriffen stehen dürfen.
- Der Satzabstandsoperator `s` gibt an, dass zwei Suchbegriffe innerhalb einer bestimmten Anzahl von Sätzen vorkommen sollen.
- Der Absatzabstandsoperator `p` gibt an, dass zwei Suchbegriffe innerhalb einer bestimmten Anzahl von Absätzen vorkommen sollen.
- Um einen Abstandsoperator zu verwenden, stellt man ihm einen Schrägstrich voran (oder ein Prozentzeichen, wenn der zweite Suchbegriff ausgeschlossen werden soll). Zusätzlich kann man mit Zahlen den minimalen und maximalen Abstand definieren sowie mit einem Plus- oder Minuszeichen vor dem Abstandsoperator die gewünschte Reihenfolge der Suchbegriffe:
  - `&Ente /w1 &Brot` sucht nach Flexionsformen der beiden Wörter *Ente* und *Brot* im Abstand von einem Wort, also direkt nacheinander (wobei dazwischen noch ein Satzzeichen stehen kann). Die Reihenfolge ist egal.
  - `&Ente /+w1 &Brot` ist im Prinzip die gleiche Anfrage, nur dass hier eine Form von *Brot* zwingend auf eine Form von *Ente* folgen muss.
  - `&Ente /-w1 &Brot` dreht die Reihenfolge um.
  - `&Ente /w4 &Brot` sucht nach den beiden Suchbegriffen im Abstand von maximal vier Wörtern (also mit maximal drei Wörtern dazwischen). Will man zusätzlich einen Mindestabstand von bspw. drei Wörtern festlegen, geht das so: `&Ente /w3:4 &Brot`. Damit lässt sich insbesondere auch ein exakter Abstand festlegen: `&Ente /w4:4 &Brot`.
  - Als Abstand lässt sich auch 0 angeben. Bei Wörtern lassen sich mit `/w0` z. B. Satzzeichen finden, die direkt hinter Wörtern stehen: `so /w0 ,` (Komma nach *so*). Außerdem dient `/w0` in wortartenannotierten Korpora dazu, nach Wörtern zu suchen, die zusätzlich einer bestimmten Wortart angehören sollen (siehe Abschnitt 3.7). Mit `/s0` und `/p0` kann man angeben, dass zwei Suchbegriffe im selben Satz bzw. Absatz vorkommen sollen.
  - Bei komplexeren Suchanfragen mit mehr als einem Abstandsoperator ist es empfehlenswert, runde Klammern zu verwenden, um Suchbegriffe zusammenzufassen: `(Was /+w1 ist) /+w5,s0 für`. `Was /+w1 ist` wird hier mit Klammern zu *einem* Suchbegriff, auf den sich dann der Abstandsoperator `/+w5,s0` bezieht.
  - Abstandsoperatoren lassen sich auch kombinieren. So gibt `/w2,s0` an, dass sich die damit verknüpften Suchbegriffe im selben Satz im Abstand von maximal zwei Wörtern befinden sollen.

Um nun z. B. nach der rheinischen Verlaufsform (*am*-Progressiv: *Ich bin noch am arbeiten.*) zu suchen, könnte man nach *am* suchen, gefolgt von einem Verb im Infinitiv, zusammen mit

1	A00/FEB.11611	...en: Wenn er nicht gestorben	ist, will sagen, wenn die Dampfmaschine nicht kaputt gegang...
2	A00/FEB.13149	...erzeit in etlichen Gemeinden	am entstehen ist.
3	A00/FEB.13283	Das Unternehmen	ist stark am wachsen: Dieses Jahr will die Firma mit 2000 An...
4	A00/APR.23481	Fast pausenlos	war er am messen und beraten - «Welche Krawatte passt am...
5	A00/MAI.35661	...n Gemeinderat Roland Heule	ist am wirken.
6	A00/JUL.48036	...ulitag anno 1943, heute aber	ist nichts damit, von einem Sommerjahrmarkt in Marbach wei...
7	A00/AUG.55835	...age nach dieser Berufsgruppe	ist am wachsen.
8	A00/SEP.60898	... an die Hantel, ab 15.30 Uhr	sind die Rorschacher Nachwuchsheber und ab 17 Uhr die Elit...
9	A00/SEP.63335	«Meine Frau und ich	waren gestern bis um elf Uhr nachts am backen; ich selbst ha...
10	A00/OKT.69309	...hnee überrascht uns, doch er	ist wieder am schmelzen.
11	A00/NOV.76031	...stätigen, dass ich noch nicht	am vergreisen bin, sondern meine Aufgaben mit Kraft und Aus...
12	A00/NOV.81377	...Präsidentenwahl in den USA	seien sie heute noch am zusammenzählen.
13	A00/DEZ.84209	...hen in diesem Bereich tüchtig	«am rotieren» sind.
14	A01/FEB.09644	...hweiz, der in Läufelfingen BL	am entstehen ist.
15	A01/OKT.34648	Die Turnerinnen und Turner	waren mit viel Eifer und Freude am ausprobieren.
16	A01/NOV.43106	Im Klassenzimmer nebenan	sind Karten am entstehen.
17	A01/DEZ.51415	...enn die Töne springen, dann	sind die Jumping Notes am swingen» werden die sechs Dixie...
18	A01/DEZ.52350	...enn die Töne springen, dann	sind die Jumping Notes am swingen», brachten die sechs ang...
19	SPK/J04.01473	...en verzerrten Gesichtspartien	sind eigentlich für Betrachter am angsteinflössensten?

Abbildung 5: Ergebnisdarstellung in COSMAS II_{web} (KWIC): *am*-Progressiv

einer Form von *sein* im selben Satz: `(am /+w1 MORPH(VRB inf)) /s0 &sein`.[11] Einige Ergebnisse sind in Abb. 5 dargestellt (inkl. eines *false positive*, denn *angsteinflössensten* wurde offenbar fälschlich als Verb getaggt). Eine solche Suche wird allerdings keine Sätze finden, in denen das Verb(?) großgeschrieben wurde, da es in solchen Fällen (höchstwahrscheinlich) als Substantiv annotiert wurde. Sollen auch solche Sätze gefunden werden, wird die Abgrenzung zu normalen Substantiven sehr schwierig (*am Wegesrand* u. ä.).

3.9 Position im Satz

Zwar lässt sich für einzelne Wörter nicht direkt deren gewünschte Position im Satz angeben, mit den Suchbegriffen `<sa>` und `<se>` lässt sich jedoch nach Satzanfang bzw. Satzende suchen (analog dazu gibt es auch noch `<pa>` für den Absatzanfang, `<pe>` für das Absatzende, `<ta>` für den Textanfang und `<te>` für das Textende). Der Begriff steht dabei für das erste bzw. letzte Wort im Satz und lässt sich mit `/w0` auch mit einem konkreten Wort verknüpfen, sodass z. B. `Linguisten /w0 <sa>` nach *Linguisten* am Satzanfang sucht. Alternativ lässt sich die Suchanfrage auch so formulieren: `Linguisten:sa` oder `#BED(Linguisten, sa)` – diese Varianten sorgen für eine etwas schnellere Suche. Die Beispiele aus Abschnitt 2.6 in COSMAS-Syntax:

- *Minister* an dritter Position im Satz: `<sa> /+w2:2 Minister`[12]

[11] Hier ist darauf zu achten, dass `&sein` auch die Formen des Possessivpronomens erfasst – diese müssen in der Wortformenliste manuell abgewählt werden, um bessere Ergebnisse zu erhalten.

[12] Stört bei einer Anfrage wie dieser, dass in den Ergebnissen auch stets das erste Wort im Satz markiert wird,

- Fragezeichen am Satzende: `\? /w0 <se>`
- ein finites Voll-, Hilfs- oder Modalverb am Satzende (in einem Korpus, das mit dem TreeTagger annotiert wurde): `MORPH(VRB fin) /w0 <se>`

3.10 Logische Verknüpfungen

COSMAS II kennt die logischen Operatoren `und`, `oder` und `nicht` (will man nach den Wörtern *und*, *oder* und *nicht* suchen, muss man Anführungszeichen setzen: `"nicht"`). Logische Operatoren beziehen sich stets auf ganze Texte.

Um nach allen Texten zu suchen, die *Linguist* und/oder *Sprachwissenschaftler* enthalten (also mindestens eines der beiden Wörter), kann man die folgende Suchanfrage verwenden: `Linguist oder Sprachwissenschaftler`. Sollen beide Wörter im selben Text enthalten sein, verwendet man `und`. Und um nach Texten zu suchen, die *Linguist* enthalten, nicht aber *Sprachwissenschaftler*, wird `nicht` vor den auszuschließenden Suchbegriff gesetzt: `Linguist nicht Sprachwissenschaftler`.

Bei Verwendung logischer Operatoren ist es oft entscheidend, Suchbegriffe mit runden Klammern richtig zusammenzufassen: `(dem oder einem) /+w1:1 Narren` ist etwas ganz anderes als `dem oder (einem /+w1:1 Narren)`. Ersteres findet Sätze, in denen *dem Narren* oder *einem Narren* vorkommt, letzteres dagegen Sätze, in denen *dem* oder *einem Narren* enthalten ist.

3.11 Reguläre Ausdrücke

Seit Juni 2015 ist es bei COSMAS II ebenfalls möglich, auf Wortebene reguläre Ausdrücke zu verwenden (siehe Abschnitt 2.8). Dazu wird der Operator `#REG()` benötigt: Mit der Abfrage `#REG( ^Stud(ent(en|[iI]nnen|in)?|ierende[rn]?)$ )` findet man z. B. Sätze mit *Student*, *Studentin*, *Studenten*, *Studentinnen*, *StudentInnen*, *Studierende*, *Studierender* oder *Studierenden*. `^` markiert hierbei den Anfang, `$` das Ende des Suchwortes (lässt man diese Begrenzungen weg, wird auch nach Wörtern gesucht, die vor oder nach dem Suchausdruck noch weitergehen).

3.12 Ergebnisbereich

Es existieren verschiedene Textbereichoperatoren, die es erlauben, in den Ergebnissen nur bestimmte Teile einer Suchanfrage anzuzeigen. Ist man bspw. nur an den Verben interessiert, die mit dem *am*-Progressiv gebraucht werden, kann man eine Anfrage wie folgende verwenden: `#RECHTS(&sein /+s0 (am /+w1 MORPH(VRB inf)))`.[13] `#RECHTS()` sorgt dafür, dass jeweils nur das letzte Wort aus der Treffergruppe als Treffer angezeigt wird (an der Menge der Treffer ändert sich dadurch nichts) – dies erlaubt es z. B., die Verben in der Ansicht nach Wort-Types nach ihrer Häufigkeit zu sortieren (siehe Abschnitt 3.3).

Neben `#RECHTS()` gibt es `#LINKS()` (Anzeige des ersten Wortes), `#INKLUSIVE()` (zusätzlich zu den Wörtern der Suchanfrage werden auch alle Wörter, die dazwischen vorkommen, in

lässt sich ein Textbereichoperator verwenden (siehe Abschnitt 3.12): `#RECHTS(<sa> /+w2:2 Minister)`. Dadurch wird nur der rechte Teil des Suchbegriffs markiert.

[13] Im Gegensatz zur Abfrage weiter oben gibt diese allerdings vor, dass die Form von *sein* zuerst kommen muss.

die Treffergruppe aufgenommen) und `#EXKLUSIVE()` (nur die Wörter zwischen den Wörtern der Suchanfrage werden als Treffer angezeigt).

3.13 Kookkurrenzanalyse

Kookkurrenz- oder Kollokationsanalysen dienen, wie in Abschnitt 2.10 beschrieben, dazu, Wörter zu finden, die auffällig oft im Umfeld eines Suchbegriffs vorkommen. Die Kookkurrenzanalyse von COSMAS II_{web} berechnet ausschließlich die *Log-Likelihood Ratio* (LLR); zur praktischen Anwendung reicht es normalerweise, zu verstehen, dass höhere LLR-Werte i. a. bemerkenswertere Kookkurrenzen anzeigen.

Zu den diversen Optionen der Kookkurrenzanalyse gibt es ein Tutorial unter http://www1.ids-mannheim.de/kl/misc/tutorial.html.

Für viele Analysen kann man die Standardwerte verwenden oder lediglich die Größe des Umfelds verändern, das berücksichtigt werden soll (Kontext: X Wörter links vom Suchbegriff, Y Wörter rechts davon). Nachdem man die Ergebnisse einer Suchanfrage erhalten hat, klickt man dazu einfach auf „Kook.“.

4 DGD

Die DGD (Datenbank für Gesprochenes Deutsch) des IdS ist unter http://dgd.ids-mannheim.de zu erreichen. Um auf die Korpora zugreifen zu können, ist eine kostenlose Registrierung erforderlich.

Eine Übersicht über die enthaltenen Korpora ist unter http://dgd.ids-mannheim.de/dgd/pragdb.dgd_extern.sys_inv zu finden. Unter den insgesamt 24 Korpora befinden sich Korpora zu Mundarten, sehr speziellen Varietäten (z. B. zum Emigrantendeutsch von Wienern in Jerusalem), aber auch zu Umgangs- und Standardsprache. Besonders hervorzuheben ist wahrscheinlich das FOLK-Korpus (Forschungs- und Lehrkorpus Gesprochenes Deutsch; http://agd.ids-mannheim.de/folk.shtml). Der Umfang der Korpora schwankt zwischen knapp 10 000 und 3 884 000 Wortformen; insgesamt sind knapp 9 Millionen Wortformen in der Datenbank enthalten.

Die Korpora enthalten ihrerseits nicht nur Transkripte von Aufnahmen, sondern auch die Audioaufnahmen selbst.

Bedauerlicherweise ist die Abfragesyntax eine andere als bei COSMAS. Die Korpora sind überdies zwar wortartenannotiert, die Annotation weist jedoch noch eine so hohe Fehlerrate auf, dass die Möglichkeit, nach Wortarten zu suchen, extra freigeschaltet werden muss.[14] Auf Anfrage kann man dazu ein Passwort erhalten.

4.1 Sucharten

Welche Korpora bei einer Suchanfrage berücksichtigt werden sollen, lässt sich auf der linken Seite auswählen. In der DGD gibt es drei Möglichkeiten, die Korpora zu durchsuchen, wobei sowohl Transkripte als auch Metadaten durchsucht werden können:

[14] Automatische Tagger wie der TreeTagger haben mit Spontansprache oft Probleme. Texte müssen vor dem Tagging normalisiert werden, was bedeutet, dass nicht-standardsprachliche Formen auf ihre standardorthographischen Entsprechungen zurückgeführt werden müssen. Außerdem ist es z. B. nicht besonders ratsam, Spontansprache mit einem Tagger zu annotieren, der anhand von Zeitungstexten trainiert wurde, und auch das Tagset muss womöglich ein anderes als für Texte geschriebener Sprache sein …

1. Volltext: Mit der Volltextsuche lassen sich leicht Dokumente finden, die bestimmte Wörter enthalten. Die Verwendung von Platz- und Abstandshaltern sowie logischen Operatoren ist möglich. Üblicherweise wird man wohl nach Wörtern in den Transkripten suchen, es ist aber auch möglich, nach Metadaten in Ereignisdokumentationen oder Sprecherdokumentationen zu suchen (z. B. nach `Lehrer%`, um Sprechereignisse von Lehrern oder Lehrerinnen zu finden).

2. Metadaten: Nach Metadaten kann man auch direkt mit der Metadatensuche suchen. Dazu wählt man einen möglichen Deskriptor (wie Land, Region, Geschlecht oder Beruf) aus und gibt an, welchen Wert dieser annehmen soll.[15] Mögliche Werte kann man sich durch einen Klick auf den Button rechts vom Eingabefeld anzeigen lassen, weitere Deskriptoren lassen sich durch Klick auf das Plus-Symbol hinzufügen, sodass man z. B. nach männlichen Sprechern (1. Deskriptor) aus einer bestimmten Region (2. Deskriptor) suchen kann. Als Ergebnis erhält man eine Liste von Transkripten, auf welche die angegebenen Bedingungen zutreffen und die man sich einzeln anzeigen lassen kann. Die gesamte Liste lässt sich auch als virtuelles Korpus speichern, indem man auf das Diskettensymbol klickt.[16]

3. Token-Suche: Die Token-Suche ist für linguistische Fragestellungen wahrscheinlich am geeignetsten. Hier lassen sich auch zuvor gespeicherte virtuelle Korpora laden (in der Korpusauswahl links) und durchsuchen.

4.2 Einfache Suche

Werden Wörter direkt eingegeben, wird in exakt dieser Form nach ihnen gesucht. Mit der Volltextsuche ist es außerdem möglich, nach Flexionsformen zu suchen, indem man den Grundformoperator `$` verwendet: `$Hund` findet neben *Hund* auch *Hunde*, *Hundes* usw. Mit der Token-Suche ist ein solcher Operator dagegen gar nicht notwendig, da es separate Eingabefelder für transkribierte Formen im Transkript, normalisierte, also standardorthographische, „korrekt“ geschriebene Formen und Lemmata gibt.

4.3 Ergebnisse speichern

Suchergebnisse der Token-Suche lassen sich durch Klick auf das Diskettensymbol speichern und zu einem späteren Zeitpunkt erneut aufrufen. Durch einen Klick auf das zweite Symbol rechts davon lassen sich die Ergebnisse (in KWIC-Form) auch als Textdatei exportieren und herunterladen.

4.4 Suche mit Platzhaltern

Bei allen Sucharten lassen sich die Platzhalter `_` und `%` verwenden. `_` steht für genau ein beliebiges Zeichen, `%` für beliebig viele (auch 0).

[15] Auch hierbei kann man einfache Suchoperatoren wie `%` oder `|` verwenden, um bspw. nach Sprechereignissen von Lehrern oder Juristen zu suchen: `Lehrer%|Jurist%`.

[16] Alternativ kann man sich unter „Browsing“ die Bestandteile der Korpora ansehen und einzelne Sprecher oder Ereignisse einem virtuellen Korpus hinzufügen.

Abbildung 6: Ausschnitt aus einem Transkript in der DGD

4.5 Suche nach Wortfolgen

Verwendet man die Volltextsuche, so steht der Abstandsoperator `NEAR()` zur Verfügung, um nach mehreren Wörtern zu suchen, die in einem bestimmten maximalen Abstand voneinander vorkommen sollen. Innerhalb der Klammern werden drei Argumente erwartet: eine Liste aller Wörter (mindestens zwei), die gesucht werden sollen (ebenfalls in runden Klammern und mit Kommata getrennt), der maximale Abstand der Wörter (wobei 0 für das direkte Aufeinanderfolgen steht) und zuletzt der Wert *true* oder *false* – *true* gibt an, dass die Reihenfolge der Wörter eingehalten werden soll, mit *false* ist diese egal. Die Abfrage `NEAR((was, $denken), 2, true)` findet z. B. alle Dokumente, in denen das Wort *was* und eine Flexionsform von *denken* mit höchstens zwei Wörtern dazwischen vorkommen, wobei *was* zuerst stehen muss.

Bei der Token-Suche ist dieser Operator nicht verwendbar. Will man hier nach Wortfolgen suchen, muss man schrittweise vorgehen: Zunächst sucht man nach einer Wortform oder einem Lemma, anschließend klickt man auf den Reiter „Kontext“ und führt eine zweite Suche durch, die sich auf die Kontexte der im ersten Schritt gefundenen Treffer bezieht. Hat man also zunächst nach dem Lemma `denken` gesucht, kann man nun nach `was` suchen und dabei zum Kontext angeben, dass sich das Wort maximal drei Tokens links von den zuvor gefundenen Flexionsformen befinden soll (Alternativen: rechts oder beidseitig). Über „Skopus“ lässt sich außerdem einstellen, ob als Kontext nur derselbe Gesprächsbeitrag, derselbe Sprecher oder das ganze Transkript in Frage kommt. Nachdem die Suchergebnisse derart

gefiltert wurden, werden die passenden Treffer hervorgehoben, die übrigen (aus dem ersten Schritt) durchgestrichen.[17] Durch einen Klick auf das kleine Mülleimersymbol kann man die ausgefilterten Ergebnisse aus der Trefferliste entfernen.

4.6 Logische Verknüpfungen

Auch hier gibt es – bei der Volltextsuche – die drei logischen Operatoren, die auch das DWDS und COSMAS II kennen; sie werden lediglich anders eingegeben: `&` (und), `|` (oder) sowie `~` (nicht). `Hund|Katze` findet Dokumente, in denen wenigstens eins der Wörter *Hund* und *Katze* vorkommt, `Hund~Katze` findet Dokumente, in denen *Hund* vorkommt, nicht aber *Katze*.

4.7 Sonstige Operatoren

Für die Volltextsuche stehen auch noch folgende Operatoren zur Verfügung:

1. `FUZZY()` findet Wörter, die ähnlich geschrieben werden wie das in Klammern angegebene. So findet etwa `FUZZY(Stern)` u. a. Treffer für *Stefan*, *stehn* und (etwas rätselhaft) *Straßen*.

2. `!` vor einem Wort gibt an, dass auch Wörter gesucht werden sollen, die ähnlich *ausgesprochen* werden. Dazu wird der Soundex-Algorithmus verwendet, der eigentlich für das Englische entwickelt wurde, aber auch für das Deutsche häufig brauchbare Ergebnisse liefert.

3. `>` ist ein Schwellenwertoperator: Damit lassen sich Dokumente finden, die den vorangehenden Suchausdruck häufiger enthalten, als der Schwellenwert angibt. `ey>20` findet z. B. alle Dokumente, in denen *ey* mehr als zwanzigmal vorkommt.

4.8 Reguläre Ausdrücke

Bei der Token-Suche ist es möglich, reguläre Ausdrücke (siehe Abschnitt 2.8) in Abfragen zu verwenden. Dazu muss im Suchfenster lediglich ein Haken vor „Reguläre Ausdrücke“ gesetzt werden. Reguläre Ausdrücke müssen nicht (wie im DWDS) gesondert markiert werden: Abfragen wie `de(r|ss)en` (*deren* oder *dessen*) oder `ä{2,6}hm?` (mindestens zwei-, maximal sechsmal *ä*, gefolgt von *h* und einem optionalen *m*) kann man direkt im Suchfeld eingeben.

5 *Precision* und *Recall* bei Korpusabfragen

Wenn man bestimmte Wörter oder Wortfolgen in einem Korpus sucht, erhält man häufig keine perfekten Ergebnisse. So kann es sein, dass unter den ausgegebenen Treffern viele sind, die gar nicht der intendierten Abfrage entsprechen – entweder weil sie zu ungenau war oder weil bspw. Wörter im Korpus falsch annotiert sind. Die *Precision* einer Abfrage ist der Anteil der gewünschten Treffer an den ausgegebenen. Wenn also drei von zehn Treffern „falsch“ sind (*false positives*), so liegt die Genauigkeit der Abfrage bei 70%.

[17] Über den Reiter „Metadaten“ lässt sich zusätzlich auch noch nach Metadaten filtern. Dies funktioniert genauso wie bei der Metadatensuche, die in Abschnitt 4.1 beschrieben wurde.

Ebenso kann es sein, dass eine Abfrage nicht alles zurückliefert, was im Korpus dem Gesuchten entspricht. Wenn im Korpus z. B. acht relevante Wortfolgen enthalten sind, die man sucht, die Abfrage aber nur sechs davon ausgibt (plus etwaige *false positives*), hat sie einen *Recall* (eine Trefferquote) von 6/8, also 75%. Die zwei nicht ausgegebenen Wortfolgen nennt man auch *false negatives* (weil sie fälschlich als nicht relevant eingestuft wurden). Problematisch ist hier natürlich, dass man normalerweise gar nicht weiß, wie viele gewünschte Wortfolgen überhaupt im Korpus enthalten sind.

Precision und *Recall* beeinflussen sich häufig gegenseitig: Eine höhere Genauigkeit zieht oft eine geringere Trefferquote nach sich, eine höhere Trefferquote umgekehrt eine geringere Genauigkeit.