Lösungshinweise zu den Aufgaben

zu Kap. 2.2.2
Die Datei *weibfrau.csv* findet sich im Ordner „Aufgaben“. Konsultieren Sie Tutorial „01-Grundlegendes“ (dort insbesondere Kap. 2 zu Excel und Calc), um die Datei in ein Tabellenkalkulationsprogramm einzulesen. Fügen Sie dann eine neue Annotationsspalte hinzu, die Sie ausfüllen; mit Hilfe von Tutorial „06-Tabellen und Grafiken mit Excel und Calc“ können Sie auch erste grundlegende Auswertungen Ihrer Annotationen vornehmen.

zu Kap. 2.2.3
1. Eine mögliche Störvariable ist, dass die Kunstwörter zum Teil große Ähnlichkeit zu existierenden Wörtern des Deutschen aufweisen; allerdings stellt sich zugleich auch die Frage, inwieweit sich dieses Problem vermeiden lässt, ohne mit Wörtern zu arbeiten, die die Eigenschaften, deren Einfluss man überprüfen möchte, gar nicht aufweisen.
2. Für Lesezeitexperimente kann man z.B. die Software DMDX verwenden, wenn man nicht selbst ein einfaches Programm z.B. mit PsychoPy oder jsPsych programmieren möchte. Allerdings zielt die Frage mehr auf ein Konzept für ein solches Experiment als auf seine technische Umsetzung ab. Denkbar wäre beispielsweise, die Hypothese zu überprüfen, dass Lesezeiten einen Hinweis darauf geben, wie etabliert eine Form ist – klar falsche Formen wie *des Matroses* sollten in diesem Fall langsamer gelesen werden, weil ProbandInnen über die Form „stolpern“, während schon etablierte Formen wie *des Schwans* in normaler Geschwindigkeit gelesen werden.

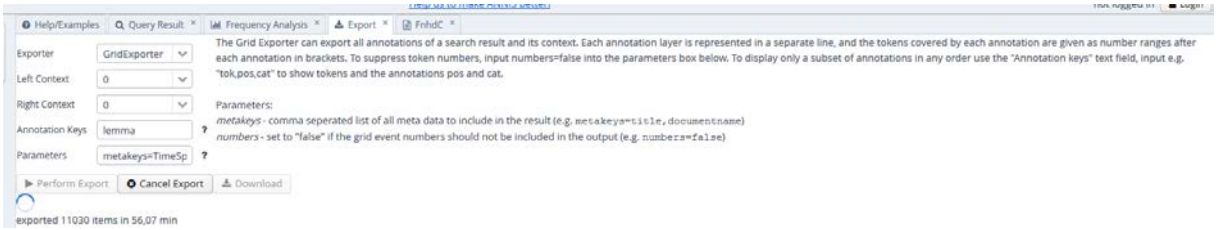
zu Kap. 4.1.1
1. Die Alternanz von /f/ und /b/ in ***f****rater* – ***B****ruder* lässt sich nicht mit der 1. Lautverschiebung erklären. Für *e**d**ere* – *e**ss**en* brauchen wir zusätzlich noch die 2. Lautverschiebung.
2. *was* – *war:* Dieser Unterschied lässt sich über das Vernersche Gesetz und den grammatischen Wechsel erklären und hat nichts mit der 2. Lautverschiebung zu tun.

zu Kap. 5.2.2
Da im Bonner Frühneuhochdeutschkorpus Suffixe eigens getaggt sind, können wir mit einer sehr simplen Suchanfrage nach Derivaten mit dem Suffix *-lich* suchen. Im ANNIS-Interface unter https://korpora.zim.uni-due.de/annis geben wir einfach ein:

```
anno:suffix="lich"
```

und erhalten alle 6078 Belege. (Hinweis: Das „anno:“-Präfix wird im FnhdC gebraucht, um zu vermeiden, dass die Ergebnisse doppelt gefunden werden, wie es auf einigen der folgenden Screenshots der Fall ist...) Allerdings helfen uns diese Rohdaten zunächst nur bedingt weiter: Wir brauchen, wie wir in Kap. 5.2.2 gesehen haben, die Anzahl der Types und Tokens und insbesondere auch der Hapax Legomena, um die potentielle Produktivität zu berechnen. Wie kommen wir an diese Zahlen?
Prinzipiell gibt es in ANNIS zwar die Möglichkeit, Frequenzlisten zu generieren und auch zu exportieren (mit dem Button „Frequency Analysis“), allerdings funktioniert das bei dieser konkreten Anfrage nicht, da es aufgrund der relativ hohen Trefferzahl zu einem Timeout kommt. Deshalb müssen wir den Umweg über den Export der Daten nehmen. Wie man Daten aus ANNIS exportiert, wird u.a. im Tutorial „03-ANNIS“ erklärt. Zwar gibt es in der ANNIS-Instanz, die FnhdC verwendet, (derzeit) keinen TextColumnExporter, mit dem man KWIC-Konkordanzen exportieren kann, doch sind wir in diesem Fall auch gar nicht auf KWIC-Export

angewiesen, da wir ja nur die Wortbildungsprodukte selbst – idealerweise: die Lemmata – exportieren möchten.
Leider funktionieren die Exporter im FnhdC derzeit nicht so, wie sie eigentlich sollten – einige geben (zumindest bei mir, in Firefox und Chrome) Fehlermeldungen aus. Man kann den GridExporter verwenden, allerdings braucht er eine gute Stunde zum Export aller Belege (mit den im Screenshot unten angegebenen Einstellungen).



Die Exportdatei sieht wie folgt aus:

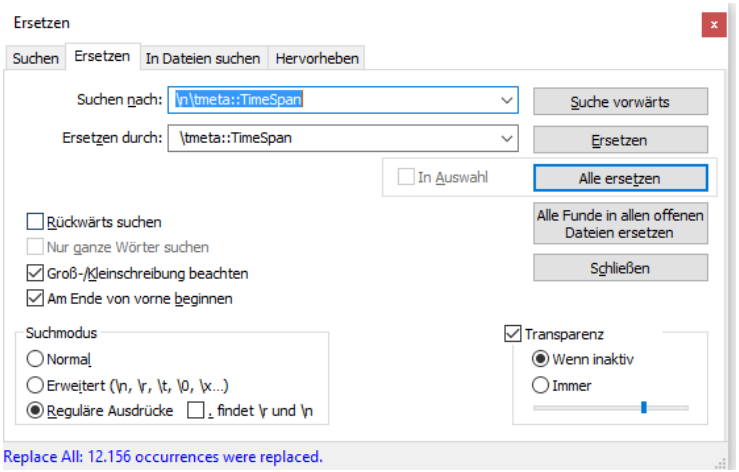
```
0.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=rechtlich">rechtlich</a>[1-1]
    meta::TimeSpan  1

1.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=rechtlich">rechtlich</a>[1-1]
    meta::TimeSpan  1

2.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=maennlich">männlich</a>[1-1]
    meta::TimeSpan  1

3.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=maennlich">männlich</a>[1-1]
    meta::TimeSpan  1

4.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=ritterlich">ritterlich</a>[1-1]
    meta::TimeSpan  1

5.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=ritterlich">ritterlich</a>[1-1]
    meta::TimeSpan  1

6.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=kaiserlich">kaiserlich</a>[1-1]
    meta::TimeSpan  1

7.  lemma    <a target="_blank" href="http://www.woerterbuchnetz.de/cgi-bin/WBNetz/startGlobalSearch.tcl?stichwort=kaiserlich">kaiserlich</a>[1-1]
    meta::TimeSpan  1

8.  lemma    geßlich[1-1]
    meta::TimeSpan  1

9.  lemma    geßlich[1-1]
    meta::TimeSpan  1

10. lemma    gänglich[1-1]
    meta::TimeSpan  1

11. lemma    gänglich[1-1]
    meta::TimeSpan  1

12. lemma    eigentlich[1-1]
    meta::TimeSpan  1

13. lemma    eigentlich[1-1]
    meta::TimeSpan  1
```

Wie Sie sehen, wurde nur die Lemma-Annotation exportiert, wie ich es auch im Screenshot oben unter „Annotation Keys“ angegeben hatte. Weiterhin habe ich noch die Zeitspanne mitexportiert, um auf Wunsch auch diachrone Auswertungen vornehmen zu können.
Diese Exportdatei muss natürlich noch bearbeitet werden, um sinnvoll in Tabellenkalkulationsprogramme eingelesen zu werden. Zum Glück ist der Aufbau relativ übersichtlich: Zuerst haben wir immer eine laufende Nummer, dann den Namen der jeweiligen Annotation („lemma“ bzw. „meta::TimeSpan“), gefolgt von einem Tabstopp und dem Wert der jeweiligen Annotation (z.B. lemma=“gänglich“ etc.).
Wir hätten das Ganze nun gern in folgendem Format:

lemma meta::TimeSpan
rechtlich 1
rechtlich 1
männlich 1
etc.

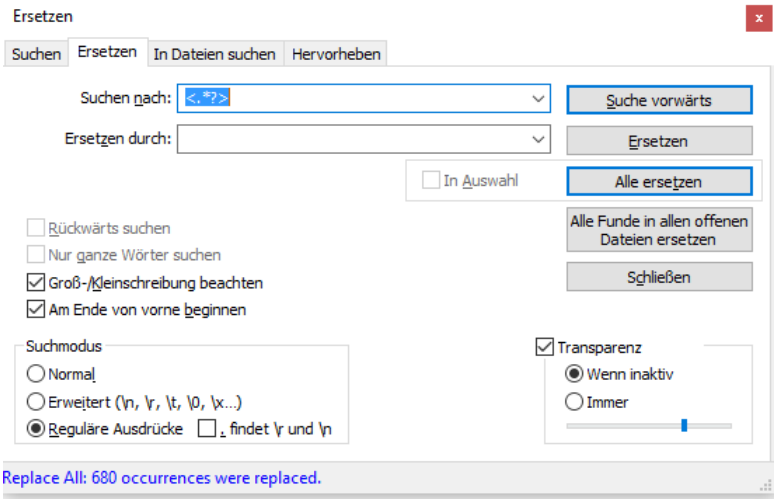
Um das zu erreichen, tun wir Folgendes:

1. Wir ersetzen \n\tMeta::TimeSpan durch \tMeta::TimeSpan (\n steht für Zeilenumbruch, \t für Tabstopp):



Dadurch werden die TimeSpan-Annotationen, die derzeit noch eine eigene Zeile beanspruchen, quasi nach oben gezogen.

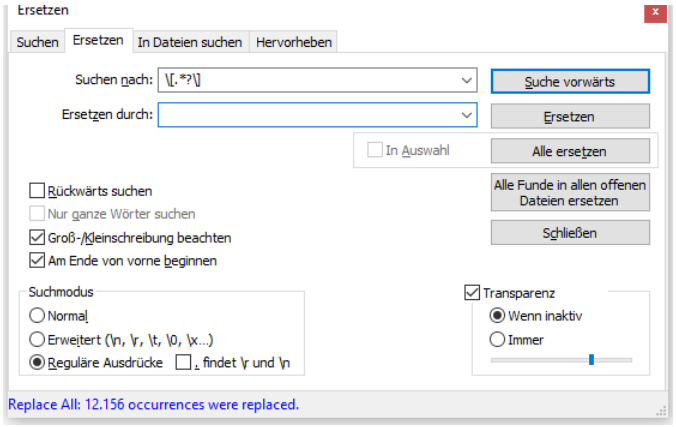
2. Um die Attribute in Klammern mit den Links zum Wörterbuchnetz loszuwerden (siehe Screenshot oben), benutzen wir folgenden Ersetzungsbefehl:



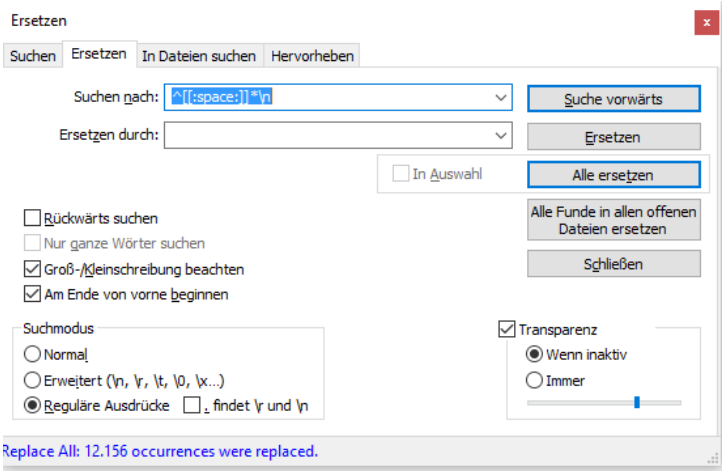
Denn wie der Screenshot zeigt, beginnen die Links jeweils mit `<a target`, und nach dem Lemma folgt noch der schließende Tag `</a>`. Das Fragezeichen wird hier benutzt, um den davor stehenden Ausdruck „non-greedy“ zu machen: Ohne das Fragezeichen würde der Ersetzungsbefehl alles ersetzen, was zwischen dem ersten < im Dokument und der letzten schließenden eckigen Klammer > im Dokument steht. Das wäre ganz schön viel Text, und vor allem wäre vieles dabei, was wir noch brauchen. In

dieser „nicht-gierigen“ Form hingegen ersetzt er genau das, was wir ersetzt haben möchten.

3. Das gleiche tun wir jetzt noch für die Elemente in eckigen Klammern, die wir nicht brauchen (obwohl wir sie theoretisch auch behalten könnten, weil immer das gleiche da steht, nämlich *[1–1]*, sodass das auf die Auswertung keinen Einfluss hat). Weil [und] als Gruppierungszeichen verwendet werden, müssen wir den Escape-Slash \ verwenden, um die „echten“ Klammern zu finden:

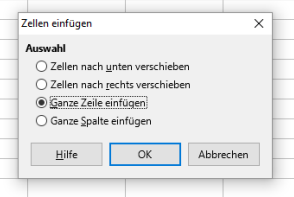


4. Zuletzt entfernen wir noch das „finished“ in der allerletzten Zeile sowie die leeren Zeilen. Das geht in Notepad++ sehr einfach unter Bearbeiten>Zeilenoperationen>Leerzeilen löschen oder aber mit dem Befehl, ^[[:space:]]*\n (also ein Zeilenumbruch, vor dem allenfalls eine beliebige Anzahl an Whitespaces steht) durch nichts zu ersetzen:



Das Ganze können wir jetzt in Excel oder Calc copy&pasten. Ich benutzte hier Calc zur Veranschaulichung.
Zunächst fügen wir eine neue Zeile vor der allerersten Zeile ein (eine Zelle in der ersten Zeile markieren > Rechtsklick > Einfügen > Ganze Zeile einfügen).

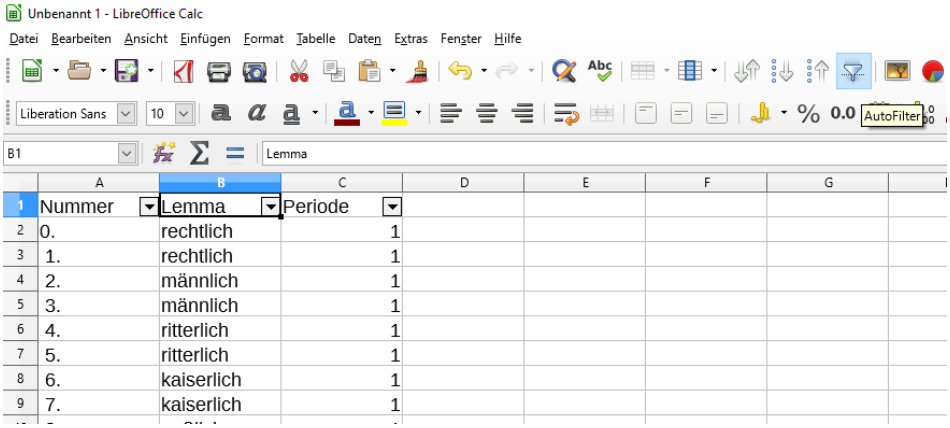
0.	lemma	rechtlich	meta::TimeSt	1
1.	lemma	rechtlich	meta::TimeSt	1
2.	lemma	männlich	meta::TimeSt	1
3.	lemma	männlich	meta::TimeSt	1
4.	lemma	ritterlich	meta::TimeSt	1
5.	lemma	ritterlich	meta::TimeSt	1
6.	lemma	kaiserlich	meta::TimeSt	1
7.	lemma	kaiserlich	meta::TimeSt	1
8.	lemma	geßlich	meta::TimeSt	1
9.	lemma	geßlich	meta::TimeSt	1
10.	lemma	gänglich	meta::TimeSt	1
11.	lemma	gänglich	meta::TimeSt	1
12.	lemma	eigentlich	meta::TimeSt	1
13.	lemma	eigentlich	meta::TimeSt	1
14.	lemma	wöhnlich	meta::TimeSt	1
15.	lemma	wöhnlich	meta::TimeSt	1
16.	lemma	eigentlich	meta::TimeSt	1
17.	lemma	eigentlich	meta::TimeSt	1
18.	lemma	sichtiglich	meta::TimeSt	1
19.	lemma	sichtiglich	meta::TimeSt	1
20.	lemma	schmählich	meta::TimeSt	1
21.	lemma	schmählich	meta::TimeSt	1
22.	lemma	wissentlich	meta::TimeSt	1
23.	lemma	wissentlich	meta::TimeSt	1
24.	lemma	völliglich	meta::TimeSt	1



Dank der neuen Zeile können wir den Spalten Namen geben:

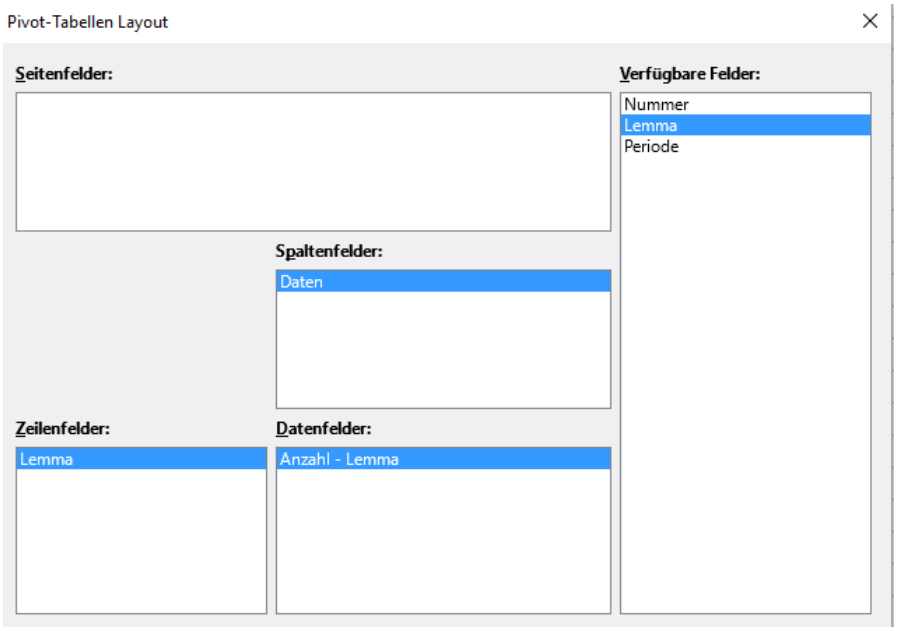
	A	B	C	D	E	F
1	Nummer	Anno1	Lemma	Anno2	Periode	
2	0.	lemma	rechtlich	meta::TimeSt	1	
3	1.	lemma	rechtlich	meta::TimeSt	1	
4	2.	lemma	männlich	meta::TimeSt	1	
5	3.	lemma	männlich	meta::TimeSt	1	
6	4.	lemma	ritterlich	meta::TimeSt	1	
7	5.	lemma	ritterlich	meta::TimeSt	1	
8	6.	lemma	kaiserlich	meta::TimeSt	1	
9	7.	lemma	kaiserlich	meta::TimeSt	1	
10	8.	lemma	geßlich	meta::TimeSt	1	
11	9.	lemma	geßlich	meta::TimeSt	1	
12	10.	lemma	gänglich	meta::TimeSt	1	
13	11.	lemma	gänglich	meta::TimeSt	1	
14	12.	lemma	eigentlich	meta::TimeSt	1	
15	13.	lemma	eigentlich	meta::TimeSt	1	
16	14.	lemma	wöhnlich	meta::TimeSt	1	
17	15.	lemma	wöhnlich	meta::TimeSt	1	

Die Spalten „Anno1“ und „Anno2“ können wir nun eigentlich löschen (durch Klick auf das B bzw. E ganz oben die ganze Spalte markieren, dann Strg+–).
In der für uns wichtigen Spalte „Lemma“ fällt auf, dass hier z.T. Leerzeichen vor den Lemmata stehen. Da das im schlimmsten Fall unserer Auswertung schaden kann, wollen wir sie loswerden. Deshalb markieren wir die gesamte Spalte, gehen dann mit Strg+Alt+F oder Bearbeiten > Suchen und Ersetzen ins Suchen-und-Ersetzen-Menü. Im Feld „Suchen“ geben wir einfach ein Leerzeichen ein, im Feld „Ersetzen“ nichts, und dann klicken wir auf „Alle ersetzen“ – voilà!
Jetzt kann es an die Auswertung gehen. Dafür setzen wir zuerst in der ersten Zeile einen Filter, indem wir eine beliebige ausgefüllte Spalte in der ersten Zeile markieren und oben auf das AutoFilter-Symbol klicken.



	A	B	C	D	E	F	G
1	Nummer	Lemma	Periode				
2	0.	rechtlich	1				
3	1.	rechtlich	1				
4	2.	männlich	1				
5	3.	männlich	1				
6	4.	ritterlich	1				
7	5.	ritterlich	1				
8	6.	kaiserlich	1				
9	7.	kaiserlich	1				

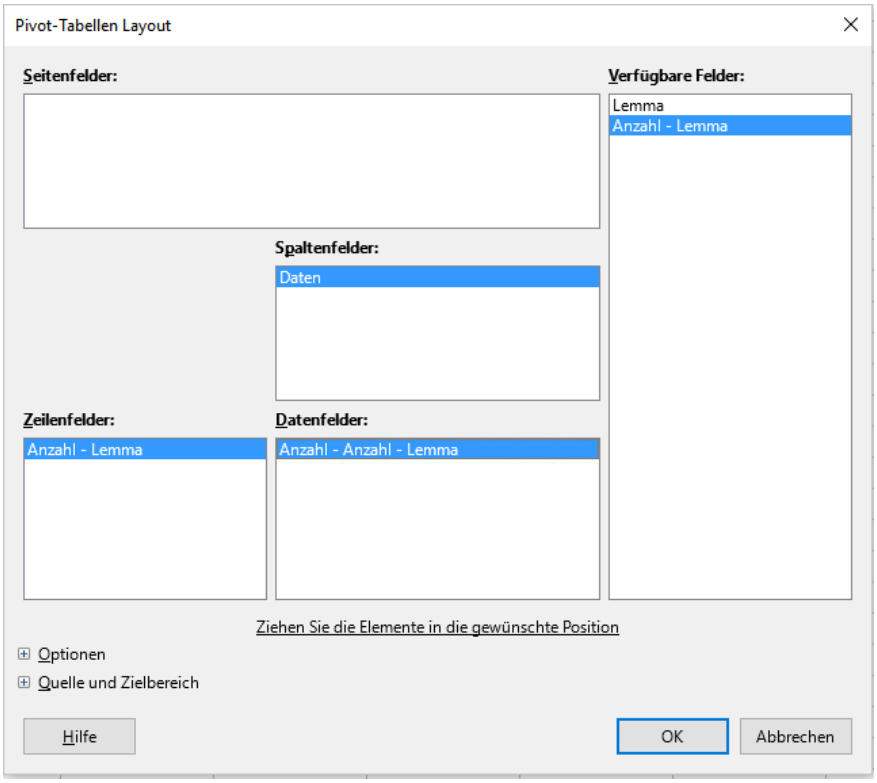
Jetzt können wir die **PivotTable**-Funktion verwenden, um die Lemmata auszuzählen: Einfügen > Pivot-Tabelle; „Lemma“ sowohl ins Fenster „Zeilenfelder“ als auch ins Fenster „Datenfelder“ ziehen, in „Datenfelder“ auf Lemma doppelklicken und die Option „Anzahl“ auswählen; jetzt sieht das Layoutfenster so aus:



Auf einem neuen Arbeitsblatt erscheint nun die Tabelle, in der die Lemmata ausgezählt werden.

Lemma	Anzahl - Lemma
abenteuerlich	6
ächtlich	6
adellich*	4
ähnlich	32
ahnlich**	2
albarlich	2
allenklich*	2
ängstlich	6
antwortlich	4
ärgerlich	20
arglich	2
ärmlich	2
ärtiglich	2
artlich	8
äußerlich	60
bärmlich	14
baulich	6
begehrlich	2

Leider kann man die „Anzahl“-Spalte nicht filtern oder sonstwie manipulieren, aber auch hier hilft ein Trick: Wir markieren die gesamte Pivot-Tabelle, kopieren sie mit Strg+C, öffnen ein neues Arbeitsblatt und fügen sie mit Strg+V ein. Nun können wir die Daten filtern (z.B. nur die Hapax Legomena anzeigen lassen), und wir können aus der eingefügten Tabelle selbst eine Pivot-Tabelle generieren, diesmal mit „Anzahl - Lemma“ in den Zeilen- und Datenfeldern:



So erhalten wir die Zahl der Hapax Legomena (175), und ganz am Ende der Pivot-Tabelle erhalten wir auch noch einmal die Gesamtzahl der Tokens, die wir ja schon aus ANNIS kennen (6078). Wenn wir nun die Anzahl der Hapaxe durch die Gesamtzahl der Tokens teilen, erhalten wir die potentielle Produktivität: 175/6078 = 0,029. Diese Zahl ist natürlich für sich genommen wenig aussagekräftig, weshalb es interessant sein kann, die Analyse noch einmal für die einzelnen Zeitspannen zu wiederholen und so einen Überblick über die diachrone Entwicklung

zu bekommen, wobei man natürlich die im Buch genannten Vorbehalte im Hinterkopf behalten muss.

zu 6.2.1
Ein Korpus, das sich zur Untersuchung der Genitivstellung prinzipiell anbietet, ist das Bonner Frühneuhochdeutschkorpus (FnhdC), da dort der Kasus jeweils annotiert ist; mit der Suchanfrage anno:kasus="genitiv" lassen sich die Genitivformen über ANNIS finden, wobei man beim Export im Grunde genau so vorgehen kann wie in der Lösung zu 5.2.2 beschrieben. Da es sehr viele Treffer gibt, kann es sinnvoll sein, eine Stichprobe zu nehmen, indem man z.B. wie im Tutorial „06-Tabellen und Grafiken mit Excel und Calc“ beschrieben eine Spalte mit Zufallszahlen hinzufügt, dann die Spalte aufsteigend oder absteigend sortiert und die ersten 500, 1000, 2000... Belege nimmt, je nachdem, wie viel man annotieren möchte.

zu 6.2.2
Nach [*kein* N, *nirgends*] lässt sich in COSMAS II am einfachsten mit der Suchanfrage `kein /+w2 nirgends` suchen. (Die Suche ist damit zwar nicht auf Substantive beschränkt, aber in diesem Fall hält sich die Anzahl der Fehltreffer in Grenzen. Ggf. kann man auch den Wortabstand noch etwas erhöhen, um neben Belegen wie *kein Krieg, nirgends* auch Belege wie *kein gerechter Krieg, nirgends* zu finden; doch scheinen diese, wenn sie überhaupt vorkommen, extrem selten zu sein).

zu 6.2.4
Ein Lösungsvorschlag für die Programmierung eines action-sentence compatibility task findet sich hier: https://github.com/hartmast/sprachgeschichte/blob/master/korpusanalysen/amprog/actionsentence.html

zu 7.1.2
Nach *blöde* kann man beispielsweise im Deutschen Textarchiv (www.deutschestextarchiv.de) mit der Lemmasuche suchen: `$l=/blöde?/g`. Optional kann man auch noch die Wortart spezifizieren: `$l=/blöde?/g with $p=/ADJ.*/`

zu 8.2.2
Das R-Skript und die Konkordanzen finden sich unter https://github.com/hartmast/sprachgeschichte/tree/master/korpusanalysen/ausrufezeichen
Die Funktion *ngrams*, die in Z. 56ff. definiert wird, hat zwei Argumente; eines davon ist *n*. Dieses Argument muss man in Z. 117/118 ändern, um z.B. Bigramme oder 4-Gramme zu bekommen; konkret muss es also

```
dng_mit <- ngrams(decow_mit, 3)
dng_ohne <- ngrams(decow_ohne, 3)
```

z.B. heißen

```
dng_mit <- ngrams(decow_mit, 2)
```

```
dng_ohne <- ngrams(decow_ohne, 2)
```

oder

```
dng_mit <- ngrams(decow_mit, 4)
dng_ohne <- ngrams(decow_ohne, 4)
```

zu 9.1
Artikel zur Rechtschreibreform finden sich z.B. auf Spiegel online oder auch (in Auszügen) im ZEIT-Archiv bei DWDS. Nicht alle sind im Volltext zugänglich, aber auch die Auszüge geben schon einen guten Eindruck von der Diskussion.

zu 9.2.1
Eine Suchanfrage für *Gott, Seele, Teufel, Baum* satzintern ist z.B. (mit Lemmasuche nach den genannten Wörtern)

```
"/[^[:punct:]]/ #0 $l=/Gott|Seele|Teufel|Baum/g"
```

Eine Suchanfrage für als Substantive getaggte, klein geschriebene Tokens im Zeitraum1400–1500 ist z.B.

```
"$w=/^[a-z].*/g with $p=/N.*/g" #less_by_date[1400,1500]
```

Für groß geschriebene entsprechend

```
"$w=/^[A-Z].*/g with $p=/N.*/g" #less_by_date[1400,1500]
```

Diese Suchanfrage kombiniert mit /[^[:punct:]]/ #0 ist leider zu komplex für das Online-Interface von DTA bzw. DWDS und führt zu Timeouts; hier müsste man die satzinitialen Belege ggf. manuell herausfiltern.

zu 9.2.2
Belege wie *Wagen* vs. *Wägen* haben eine Levenshtein-Distanz von 1, d.h. man könnte gezielt nach ihnen suchen, indem man für alle Tokens im Korpus die Levenshein-Distanz errechnet und sich dann diejenigen mit der Distanz von 1 errechnet. Wenn man aber wirklich jedes Token mit jedem anderen vergleichen wollte, wäre das extrem rechenintensiv; sinnvoll ist daher z.B., nur Wörter mit gleicher Anzahl an Buchstaben zu vergleichen (in R bekommt man diese Information mit der Funktion *nchar()*) und/oder, wie in dem im Buch dargestellten Beispiel, die Treffer zunächst alphabetisch zu sortieren und dann jeden Treffer mit dem darauffolgenden zu vergleichen. Natürlich ist auch zu beachten, dass diese Methode lediglich helfen kann, Zweifelsfälle zu finden, und zwangsläufig sehr viele Fehltreffer generiert, da z.B. auch Wörter mit Tippfehlern etc. gefunden werden.